



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

ALGORITMO DE ANÁLISIS DE SIMILITUD DE IMÁGENES

TESIS DE GRADO

DIRECTOR:
NAIOUF, MARCELO

CO-DIRECTORA:
SANZ, CECILIA

AUTORES:
DE GIUSTI, LAURA
TARRÍO, DIEGO

DICIEMBRE DE 2000

FACULTAD DE INFORMÁTICA - UNIVERSIDAD NACIONAL DE LA PLATA

TES
00/4
DIF-02098
SALA



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA
Biblioteca
50 y 120 La Plata
catalogo.info.unlp.edu.ar
biblioteca@info.unlp.edu.ar



DIF-02098

DONACION.....
\$.....
Fecha..... 4-10-05
Inv. E..... Inv. B. 2098

TES
0014



AGRADECIMIENTOS

A Santy, el amor de mi vida.

A mi familia, que siempre me dio su apoyo incondicional. En especial a mi hermana Vero que gracias a sus cartitas llegué a estar viviendo este momento.

A la familia de Santy.

A mis amigas Maru, Cari y Pao.

Laury.

A Vero, que me bancó los (no escasos) momentos de mal humor.

A mi familia, a la familia de Vero y amigos.

Al negrito (mi perro).

Diego.

A todos los integrantes del Lidi (Laboratorio con Integrantes De características Inimitables).

A nuestros directores Marcelo y Ceci.

Un agradecimiento muy especial a Ceci, que nos ayudó y alimentó con golosinas durante todo el trabajo.

A nuestros compañeros más cercanos: Andrés (Linux & C master) y Seba (compañero de almuerzos y tesis).

A Tito (aunque a veces quiere echarnos) (Aguante Racing!).

Laury y Diego.



DEDICATORIAS

A Santy, el amor de mi vida.

A mi familia, en especial a mi papá.

A mis cuatro abuelos.

Laury.

A Vero.

A mis abuelas.

Diego.

**INDICE**

CONTENIDO	PÁGINA
1.INTRODUCCIÓN.....	1
1.1.PRESENTACIÓN DEL PROBLEMA.....	1
1.2.MOTIVACIÓN Y OBJETIVOS.....	4
1.3.MATERIALES Y HERRAMIENTAS.....	6
2.SIMILITUD DE IMÁGENES.....	8
2.1.CONCEPTOS DEL PROCESAMIENTO DE IMÁGENES.....	8
2.1.1REPRESENTACIÓN DIGITAL DE IMÁGENES.....	8
2.1.2.ETAPAS FUNDAMENTALES DEL PROCESAMIENTO DE IMÁGENES.....	8
2.2.PROBLEMA DE SIMILITUD DE IMÁGENES.....	10
2.2.1.CARACTERÍSTICAS DE UNA IMAGEN.....	10
2.2.2.SIMILITUD ENTRE IMÁGENES.....	13
2.2.3.CONSULTAS POR CONTENIDO.....	15
2.3.APLICACIONES DEL PROCESAMIENTO DE IMÁGENES.....	16
2.4.EXTRACCIÓN DE CARACTERÍSTICAS, CODIFICACIÓN, FIRMA DIGITAL.....	17
2.4.1.NECESIDAD DE LA CODIFICACIÓN.....	17
2.4.2.ANÁLISIS DE CARACTERÍSTICAS DE UNA IMAGEN MEDIANTE SU FIRMA DIGITAL.....	18
3.CODIFICACIÓN DE IMÁGENES DIGITALES UTILIZANDO LA TRANSFORMADA DE WAVELET.....	19
3.1.INTRODUCCIÓN.....	19
3.2.CODIFICACIÓN CON WAVELET.....	20
3.2.1.TRANSFORMADA WAVELET DISCRETA.....	20
3.2.2.VENTAJAS DE WAVELET PARA EXTRACCIÓN DE CARACTERÍSTICAS.....	21
3.2.3.APLICACIONES.....	21
3.3.HAAR WAVELET 1D, 2D.....	22
3.3.1.HAAR WAVELET 1D, 2D.....	22
3.3.2.HAAR WAVELET EN UNA DIMENSIÓN.....	22
3.3.3.HAAR WAVELET EN DOS DIMENSIONES.....	24
3.3.4.NORMALIZACIÓN.....	26
3.4.WAVELET COMO FIRMA DIGITAL PARA EL ANÁLISIS DE SIMILITUD DE IMÁGENES.....	27
3.4.1.CALCULO DE LA FIRMA DIGITAL UTILIZANDO LA TRANSFORMADA DE WAVELET.....	27
3.4.2.RECUPERACIÓN DE AL IMAGEN MEDIANTE LA TRANSFORMADA INVERSA DE WAVELET.....	30



4.MÚLTIPLES WAVELETS COMO FIRMA DIGITAL.....	33
4.1.PROBLEMA DE ESCALADO Y TRASLACIÓN.....	33
4.1.1.PROBLEMAS DE LA FIRMA DIGITAL DE UNA IMAGEN MEDIANTE WAVELET.....	36
4.2.CÁLCULO DE WAVELET CON VENTANAS DESLIZANTES DE TAMAÑO VARIABLE.....	38
4.3.EXTRACCIÓN DE REGIONES Y SUS FIRMAS DIGITALES.....	39
4.3.1.ESTRATEGIA DE CLUSTERING.....	39
4.3.2.NECESIDADES DE ALMACENAMIENTO PARA LAS MÚLTIPLES FIRMAS DIGITALES.....	40
4.3.3.ESTRUCTURA DE CLUSTERING.....	41
4.3.4.CONDICIONES DE SIMILITUD ENTRE IMÁGENES.....	41
4.4.ALGORITMO DINÁMICO PARA OBTENER LAS MÚLTIPLES FIRMAS DIGITALES DE UNA IMAGEN.....	42
4.4.1.EVALUACIÓN DE WAVELETS UTILIZANDO WAVELETS DE TAMAÑO INFERIOR.....	42
4.4.2.ALGORITMO DINÁMICO.....	42
5.PROPUESADE PARALELIZACIÓN.....	44
5.1.INTRODUCCIÓN A ALGORITMOS Y ARQUITECTURAS PARALELAS.....	44
5.2.MOTIVACIÓN DE PARALELIZACIÓN. PARALELIZACIÓN DE DATOS Y FUNCIONAL..	49
5.3.ARQUITECTURA Y ALGORITMO DE PARALELIZACIÓN.....	50
5.3.1.TOPOLOGÍA PROPUESTA DE LA ARQUITECTURA PARALELA.....	50
5.3.2.ALGORITMO PARALELO.....	51
6.RESULTADOS.....	53
6.1.IMÁGENES ESTUDIADAS.....	53
6.2.PARÁMETROS DEL ALGORITMO.....	53
6.3.RESULTADOS OBTENIDOS.....	54
7.CONCLUSIONES Y LÍNEAS DE TRABAJOS FUTURAS.....	72
7.1.CONCLUSIONES.....	72
7.2.LÍNEAS DE TRABAJOS FUTURAS.....	72



CAPÍTULO 1: INTRODUCCIÓN

1.1.PRESENTACIÓN DEL PROBLEMA

En el campo del procesamiento de imágenes, resulta de gran interés encontrar una solución optima al problema de reconocimiento de imágenes[Nat99], motivando una continua evolución en este aspecto. Como resultado se puede observar en los últimos años un gran esfuerzo de modernización en conceptos tales como extracción de características para el posterior análisis de imágenes.

El problema de análisis de similitud de imágenes se encuentra en el área de análisis de reconocimiento por patrones[Gon96], donde la computadora procesa una imagen tratando de separar objetos distintos, los cuales identificará y clasificará mediante la extracción de características y un manejo formal/heurístico de las mismas.

El problema de encontrar las imágenes que según la información pictórica que para la interpretación humana resultan similares, posee un amplio espectro de aplicaciones[Cas95][Nat99][Zha96], donde se pueden citar las librerías digitales con problemas de consulta de imágenes por contenido en bases de datos, aplicaciones médicas, servicios meteorológicos, investigaciones biológicas, servicios de noticias, procesamiento de videos, etc.

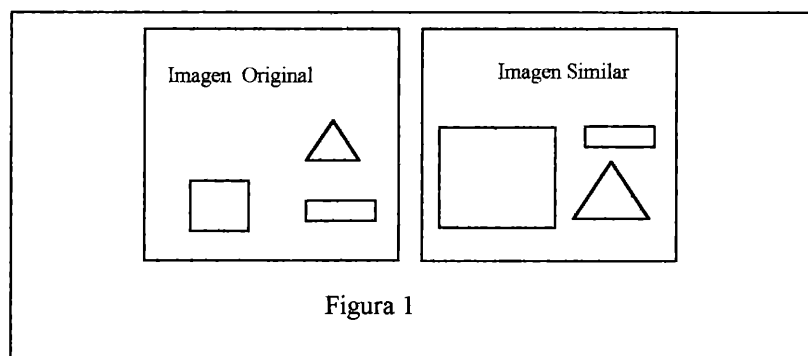
En la mayoría de los casos, estas aplicaciones necesitan resultados en tiempo real o con tiempos de respuesta muy breves. Pero como se tratará en este trabajo, los problemas de análisis de similitud entre imágenes, utilizan algoritmos con costos computacionales elevados, motivando al estudio de una posible paralelización de la solución[Hus91].

Al introducir el problema de extracción de características de una imagen se pueden citar como punto de partida los histogramas[Cra97][Gon96], usados para mostrar la distribución de los niveles de color en la imagen. Los histogramas resultan útiles para tareas de mejorado de la imagen a través de una equalización, filtrado y codificación.

La segmentación de una imagen es otro de los métodos de extracción de características, ya que se trata de una forma de clasificación de pixels, donde se asigna cada pixel en una clase basándose en alguna propiedad de los mismos[Pit93][Gon96]. La segmentación puede contribuir en el análisis de una imagen y se puede dividir en dos categorías: basadas en similitud de pixels; y basadas en discontinuidad de superficies.

Como resultado de la segmentación, las regiones “clusters” deben ser homogéneas con respecto al criterio de segmentación utilizado, como uniformidad en el nivel de color o textura; las regiones deben ser simples, no deben tener demasiados “agujeros” pequeños, y las regiones adyacentes debieran ser significativamente diferentes. El problema que presenta la segmentación es que en general se desarrolla en una base heurística específica de un tipo de aplicación determinada , dado que no existe una teoría definida para la estrategia de clustering. Una segmentación puede presentar dos tipos de errores: Generación de regiones extra , u omisión de regiones. Este problema implica que en el análisis de similitud de imágenes se pueden producir errores al concluir si una imagen es o no similar a otra.

Teniendo en cuenta que los enfoques tradicionales para consultas de imágenes basadas en contenidos utilizan una única firma digital para cada imagen haciendo uso de histogramas de color, textura, transformadas de Wavelet, etc. Estos enfoques suelen tener resultados pobres cuando las imágenes consultadas contienen objetos similares con tamaños diferentes (escalados), en lugares diferentes (trasladados), o sólo ciertas regiones de las imágenes hacen matching. Figura 1.



Como solución, se propone un algoritmo de recuperación de niveles de similaridad entre imágenes robusto a los problemas de escalado y traslación de objetos dentro de una imagen[Nat99][Nai00].

Este trabajo utiliza un modelo de análisis de similitud donde cada imagen se descompone en regiones, y las medidas de similitud entre pares de imágenes se definen en términos de la fracción del área de las dos imágenes cubiertas por regiones que hacen “matching por similitud”.

Para extraer las regiones de una imagen, se consideran ventanas deslizantes de tamaños variables y luego se agrupan por medio de una estrategia de clustering basándose en la proximidad de sus firmas (signaturas)[Zha96]. Para calcular de manera eficiente las transformadas de Wavelet en las ventanas de tamaño variable, se utiliza un algoritmo dinámico usado para calcular las firmas digitales de Wavelet, basándose en los resultados de las firmas digitales de Wavelet de tamaño inferior.

Este trabajo se aboca al análisis y prueba de una solución secuencial. Además propone un esquema de algoritmo y arquitectura para una solución paralela al problema de análisis de similitud de imágenes, utilizando como firma digital de las mismas la transformada de Wavelet en lugar de otras transformadas clásicas debido a una serie de

características deseables de dicha transformada para el tipo de problema planteado, donde resulta imprescindible extraer la mayor cantidad posible de características de la imagen como una única firma digital para su posterior comparación con otra imagen a través de esta firma.

La localidad e independencia de procesamiento entre las múltiples ventanas deslizantes y el tipo de algoritmo dinámico secuencial utilizado en este trabajo, permiten plantear una propuesta de paralelización basándose en una paralelización funcional y de datos [Kum94][Lei92][Hus91], distribuyendo fragmentos de la imagen a una arquitectura paralela con topología de árbol para extraer la codificación de la imagen utilizando múltiples Wavelets como firma digital de la misma.

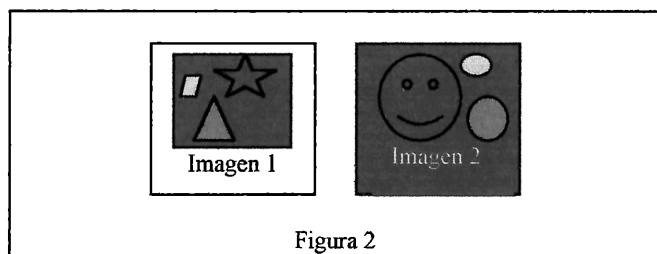
1.2.MOTIVACIÓN Y OBJETIVOS

Tradicionalmente, el problema de consultas por contenido, es decir, el problema de recuperación de imágenes que resulten similares a una imagen dada de una base de imágenes fue solucionado utilizando una firma digital para cada imagen que extraiga las características de interés para comparar. Posteriormente, utilizando una función de distancia (como métrica de diferenciación) adecuada, se puede determinar si dos imágenes poseen firmas digitales similares[Jac95].

Los métodos típicos para computar las firmas digitales fallan cuando consideran un subconjunto reducido de características de una imagen, retornando como similares imágenes que tienen ciertas características parecidas pero que no lo son considerando otras características importantes.

A modo de ejemplo, se citan las firmas digitales que incluyen los histogramas de color, que pueden utilizarse para caracterizar la composición del color de una imagen, pero no tiene en cuenta información de formas, ubicaciones ni texturas. Como resultado, dos

imágenes con composición de colores similares pueden ser totalmente diferentes en cuanto a formas y texturas. Figura 2.



Si se calcula el histograma de cada una de las imágenes presentadas en la Figura 2, y luego se realiza un análisis de similitud entre ambas utilizando como firma de cada imagen el resultado obtenido en su histograma es muy probable que la conclusión que se obtenga es que ambas imágenes son similares, aunque esto no es real. Esto se debe a que, el cálculo por histograma solo tiene en cuenta los colores de la imagen, descartando atributos de forma y textura. Como se puede observar en el ejemplo previo, los colores en ambas imágenes son coincidentes aunque las mismas no son similares.

Un enfoque para solucionar este problema es definir funciones de distancia separadas para color, forma y textura, y consecuentemente combinarlas para derivar un resultado final. El enfoque alternativo propuesto en[Wan98][Jac95] consiste en utilizar los coeficientes de Wavelet para una imagen como su firma digital. Dado que las Wavelets capturan información de forma, textura y ubicación en una única firma digital, su uso aminora algunos problemas con respecto a la utilización de los algoritmos previos.

La desventaja de este esquema de utilización de Wavelet como firma digital es que computa una única firma para la imagen entera. Como resultado, este método falla cuando las imágenes contienen objetos similares, pero en diferentes ubicaciones o cambios de tamaño[Nai00][Nat99]. Este problema se debe a que una única firma calculada para la

imagen completa no puede capturar adecuadamente las propiedades de objetos individuales.

Este trabajo utiliza un sistema robusto con respecto a cambios de resolución, efectos de sfumado (dithering), cambios de color, orientación, tamaño y ubicación, no sólo sobre la imagen completa, también sobre sus objetos individuales.

1.3.MATERIALES Y HERRAMIENTAS

En el ambiente de ejecución del trabajo se pueden encontrar dos tipos de componentes: hardware y software.

Entre los componentes de hardware podemos citar:

❖ La implementación del algoritmo de análisis de similitud entre imágenes y la propuesta de paralelización se realizaron en una arquitectura paralela que consiste de 32 procesadores transputers T805 [Tra90], que disponen de memoria local (4 MB). Además cada procesador posee cuatro links bidireccionales para comunicarse con los procesadores “vecinos”.

El algoritmo secuencial utiliza sólo uno de los transputers, a los efectos de permitir una evaluación de performance con una futura implementación paralela que haga uso de varios procesadores.

Por otro lado entre los componentes de software podemos citar:

❖ El lenguaje utilizado es C, con extensiones para los protocolos de comunicación en la arquitectura paralela.

❖ Los tipos de imágenes utilizados en este trabajo consisten en archivos bitmaps BMP [Kay95] con dimensiones $N \times N$ parametrizables (las restricciones de igual tamaño y formato cuadrado, se consideraron para permitir abocarse simplemente al problema de análisis de similitud, el trabajo puede ser extendido para funcionar con imágenes de distintos tamaños y de formatos no cuadrados).

❖ Se utilizó el espacio de color RGB para considerar los atributos de color por su amplia difusión y simplicidad de tratamiento. Para analizar la diferencia entre dos colores, se considera su distancia euclídea. Existen otras propuestas que utilizan otros espacios de color que consisten en simples transformaciones lineales del RGB, por ejemplo: YCC y CMY [Gon96][Pit93].

❖ Se utilizó el lenguaje Delphi 5.0 para realizar la interface del sistema. Esta interface es el nexo entre las tareas escritas en el lenguaje C, las cuales se ejecutan en los transputers y la pantalla que se encarga de la configuración y la elección de las imágenes para comparar. El compartimiento de datos se realiza a través de un archivo de configuración, donde se almacenan los parámetros establecidos desde la interfaz por el usuario.

CAPÍTULO 2: SIMILITUD DE IMÁGENES

2.1. CONCEPTOS DEL PROCESAMIENTO DE IMÁGENES

2.1.1. REPRESENTACIÓN DIGITAL DE IMÁGENES:

El término imagen se refiere a una función bidimensional de intensidad de luz $f(x,y)$ [Gon96], donde x e y representan las coordenadas espaciales y el valor de f en un punto cualquiera (x,y) es proporcional al brillo (ó nivel de gris) de la imagen en ese punto.

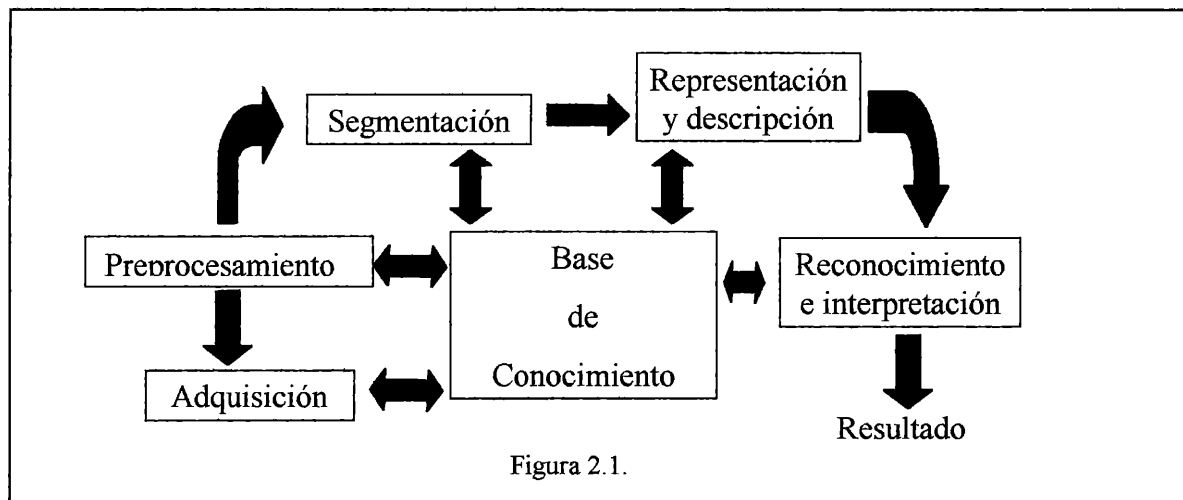
Las imágenes pueden ser: **monocromáticas** donde la función $f(x,y)$ representa el nivel de gris de la imagen en ese punto; ó **con color** donde la función representa el nivel de brillo que la misma posee en el punto x, y de la imagen.

Una imagen digital es una imagen $f(x,y)$ que se ha discretizado tanto en las coordenadas espaciales como en el brillo. La misma puede considerarse una matriz cuyos índices de fila y columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris (ó brillo) en ese punto. Los elementos de una distribución digital se denominan elementos de la imagen (pixels)[Gon96].

2.1.2. ETAPAS FUNDAMENTALES DEL PROCESAMIENTO DE IMÁGENES:

El procesamiento de imágenes, en términos generales, se refiere a la manipulación y análisis de información gráfica. Cualquier operación que actúe para mejorar, corregir, codificar, analizar, o de alguna forma cambiar la representación obtenida de una imagen se denomina “procesamiento de imágenes”.

El procesamiento de imágenes se descompone en las siguientes etapas [Gon96]: adquisición, preprocesamiento, segmentación, descripción e interpretación. (Figura 2.1).



La primera etapa del proceso es la **adquisición de la imagen**, es decir, la adquisición de una imagen digital. Para ello se puede utilizar un sensor de imágenes para una posterior digitalización de la señal producida por el mismo.

Dicho sensor puede ser una cámara de televisión, monocroma o de color, una cámara de barrido de líneas que produzca una línea de la imagen cada vez. Si la salida de la cámara o de otro sensor de imágenes no está todavía en forma digital, puede emplearse un convertidor analógico-digital para digitalizarla.

Una vez obtenida la imagen digital, la siguiente etapa trata el **preprocesamiento** de esa imagen. La función básica del preprocesamiento es la de mejorar la imagen de forma que se aumenten las posibilidades de éxito de los procesos posteriores. El preprocesamiento trata técnicas para eliminar el ruido, mejorar brillo, contraste, etc.

En la siguiente etapa se trata el problema de **segmentación** de la imagen, que consiste en partir una imagen de entrada en sus partes constituyentes u objetos. En general, es una de las tareas más difíciles en el procesamiento de imágenes, ya que la calidad con que se

efectúe esta etapa es directamente proporcional a la posibilidad de poder identificar, definir, y caracterizar a los objetos de la imagen.

A la salida del proceso de segmentación se tienen los datos de pixel en bruto, que constituyen el contorno de una región o bien todos los puntos de una región determinada. En ese caso es necesario convertir los datos a una forma adecuada para el procesamiento por computadora. Para esto existen técnicas de **representación de un contorno** (para destacar las formas exteriores) o de **representación regional** (para destacar las propiedades internas de la imagen). Algunas aplicaciones coexisten entre ambas técnicas.

También en esta etapa se pueden aplicar las técnicas de **descripción**, que consisten en extraer rasgos con alguna información cuantitativa de interés o que sean fundamentales para diferenciar una clase de objetos de otra.

2.2. PROBLEMA DE SIMILITUD DE IMÁGENES. APLICACIONES.

2.2.1. CARACTERÍSTICAS DE UNA IMAGEN:

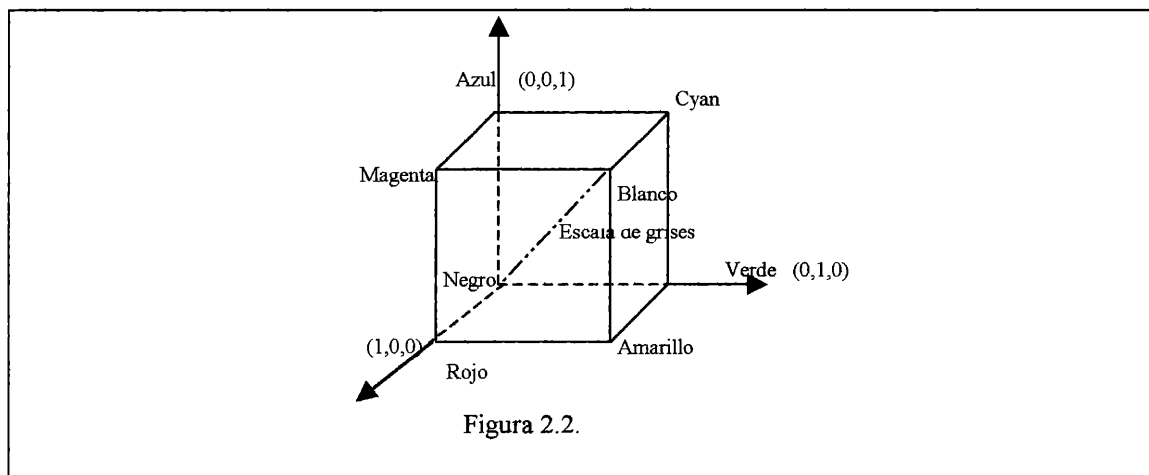
Las características de una imagen permiten describir y analizar a la misma, obteniendo información para identificarla. Entre las características de una imagen se pueden citar color, textura, forma, objetos, histograma, etc.[Pit95][Gon96][Cra97].

A continuación se presenta una breve descripción de los atributos de interés al tratar el problema de similitud entre imágenes a través de sus características:

- ❖ **Color:** el color representa un potente descriptor que a menudo simplifica la identificación de un objeto y su extracción de una imagen. El espacio de color es una forma de representar colores y sus relaciones entre cada uno. Básicamente, los colores que los seres humanos percibimos en un objeto están determinados por la naturaleza de

la luz reflejada por el objeto. Diferentes imágenes pueden utilizar distintos modelos de colores. Los monitores generalmente utilizan el modelo RGB, que consiste en representar cualquier color con los niveles de las tres componentes espectrales primarias que lo componen (Rojo Verde y Azul). Los sistemas que necesitan manipular saturación, intensidad y cromaticidad en forma separada utilizan el modelo HSI (Tono, Saturación e Intensidad). Otros modelos existentes se detallan a continuación[Hus91]:

- ✓ CMY: El subespacio de color del RGB define un cubo con los 3 ejes en los colores primarios Rojo Verde y Azul. Se puede considerar los 3 ejes restantes para representar cualquier color mediante los colores Cian, Magenta y Amarillo, que representan los colores secundarios de luz, o bien los colores primarios de los pigmentos. Figura 2.2.



- ✓ YCC: En este espacio de conversión de color, se utiliza un estándar de luminosidad/cromaticidad para imágenes de foto. Se obtiene a través de una transformación lineal del RGB, que produce una señal de luminancia y un par de señales de crominancia. La señal de luminancia converge en niveles de brillo de color. Las dos señales de crominancia convergen en la saturación y el tono del color.

- ✓ XYZ: Propuesto por la CIE (Comission International de l'Eclairage 1931) para la representación de todos los colores conocidos.
 - ✓ YIQ: Este modelo se utiliza principalmente en las emisiones comerciales de televisión. Básicamente el YIQ es una recodificación del RGB utilizada por su eficacia en la transmisión y para mantener la compatibilidad con los estándares de televisión en blanco y negro [Gon96]. Su principal ventaja es que la luminancia Y y la información de color IQ están desacopladas, permitiendo procesar la luminosidad de la imagen sin afectar su cromaticidad.
-
- ❖ Textura: esta característica es muy utilizada cuando se realiza segmentación de la imagen por regiones. Es difícil definir el concepto de textura, se puede decir que la textura proporciona intuitivamente medidas de propiedades tales como suavizado, rugosidad y regularidad. Las técnicas de descripción de texturas pueden agruparse en tres grandes clases: estadísticas (basadas en las extensiones y momentos de los histogramas por regiones, generalmente notan contrastes, suavidad, rigurosidad), espectrales (basadas en las propiedades del espectro de Fourier y se utilizan para detectar la periodicidad global de una imagen) y estructurales (describen la textura utilizando patrones primitivos acompañados de ciertas reglas de localización).
 - ❖ Forma: las formas geométricas poseen ciertas características que contiene suficiente información para aplicaciones de reconocimiento de objetos. Las características de forma pueden ser agrupadas en dos grandes clases: características de límites y características de regiones.
 - ❖ Objetos: los objetos de una imagen son los elementos que se pueden identificar en la misma después de aplicarle algún procesamiento a la imagen. Los objetos pueden ser distinguidos entre sí por su forma, textura y color.
 - ❖ Histograma: el histograma de una imagen contiene importante información sobre el contenido de la imagen. Si contiene valores de pixels concentrados en una baja

intensidad entonces se trata de una imagen oscura. Una imagen clara contiene los pixels del histograma concentrados en una alta densidad. El histograma de una imagen permite mejorar la misma distribuyendo los niveles de color para mejorar el contraste mediante una ecualización del histograma.

2.2.2. SIMILITUD ENTRE IMÁGENES:

Antes de dar una definición formal de similitud entre dos imágenes, se tiene en cuenta que, para cada imagen se obtienen las firmas sobre cada una de las regiones de la imagen (cada región es una porción de la imagen) utilizando ventanas deslizantes aplicadas a la imagen[Nat99][Nai00].

Posteriormente, se agrupan estos resultados generando una estructura que permita almacenar los resultados de manera compacta, es decir, guardando solo el centroide (promedio entre las firmas) de las firmas calculadas anteriormente cuando las mismas sean similares.

Un concepto más a tener en cuenta es la métrica de distancia que se utiliza en este trabajo. Esta métrica es la **distancia Euclídea**, la cual establece la distancia que existe entre los centroides calculados previamente de las dos imágenes a comparar.

Ahora se definen los conceptos de similitud de regiones y similitud entre un conjunto de regiones para luego dar la definición de similitud entre un par de imágenes.

Dadas dos imágenes Q y T las cuales consisten en un conjunto de regiones Q_1, \dots, Q_m y T_1, \dots, T_m respectivamente, se definen los siguientes conceptos:

Similitud de regiones: un par de regiones se considera similar si una de sus firmas cae en una distancia ϵ de la otra.

Similitud de conjunto de pares de regiones: para imágenes Q y T , y el conjunto de pares de regiones $\{(Q_1, T_1), \dots, (Q_k, T_k)\}$ se dice que es un conjunto de pares similares para Q y T, si Q_i es similar a T_i , teniendo en cuenta $i \diamond j$, $Q_i \diamond Q_j$ y $T_i \diamond T_j$ (no se repiten las regiones para Q y T).

Similitud de imágenes: se dice que dos imágenes Q y T son similares si existe un conjunto de pares de regiones similares para Q y T $\{(Q_1, T_1), \dots, (Q_k, T_k)\}$ tal que:

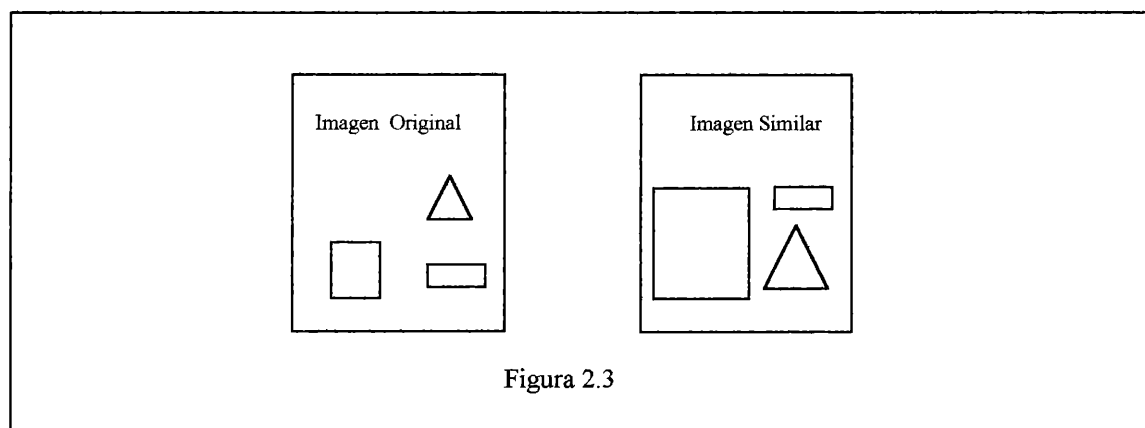
$$\text{área}(\cup Q_i) + \text{área}(\cup T_i) / \text{área}(Q) + \text{área}(T) \Rightarrow \alpha \ (\forall i = 1..k)$$

De la definición anterior se puede deducir que dos imágenes se consideran similares si la fracción del área que hace "matching" comparada con el total de áreas de las dos imágenes es mayor a un parámetro α (α es el error permitido entre dos imágenes). Hay que tener en cuenta que, permitiendo variaciones del parámetro α , se puede afectar la confiabilidad en el resultado obtenido para informar si dos imágenes son similares.

Los métodos tradicionales para calcular firmas, generalmente se basan en *algún* atributo de la imagen (por ejemplo el histograma de colores, el reconocimiento de un patrón fijo, el número de componentes de un tipo dado, etc). Esta "linealidad" de la firma hace muy difícil obtener datos sobre atributos no tenidos en cuenta en el cálculo de la firma (y que pueden afectar la similitud o diferencia entre dos imágenes).

Por ejemplo, basándose sólo en el histograma de colores, no se considera la textura, forma u objetos de la imagen, de esta manera el análisis de similitud puede fallar al determinar que dos imágenes son similares por el simple hecho que la distribución de colores son parecidas, incluso si sus objetos y/o texturas son totalmente diferentes.

Asimismo, si se considera como firma de cada una de las imágenes, (que se muestran en la siguiente figura), los objetos que las mismas contienen junto con su ubicación y tamaño, la métrica de similitud de imágenes podría fallar o no reconocer que dos imágenes son similares cuando contienen los mismos objetos los cuales se encuentran cambiados de tamaño o trasladados. Figura 2.3.



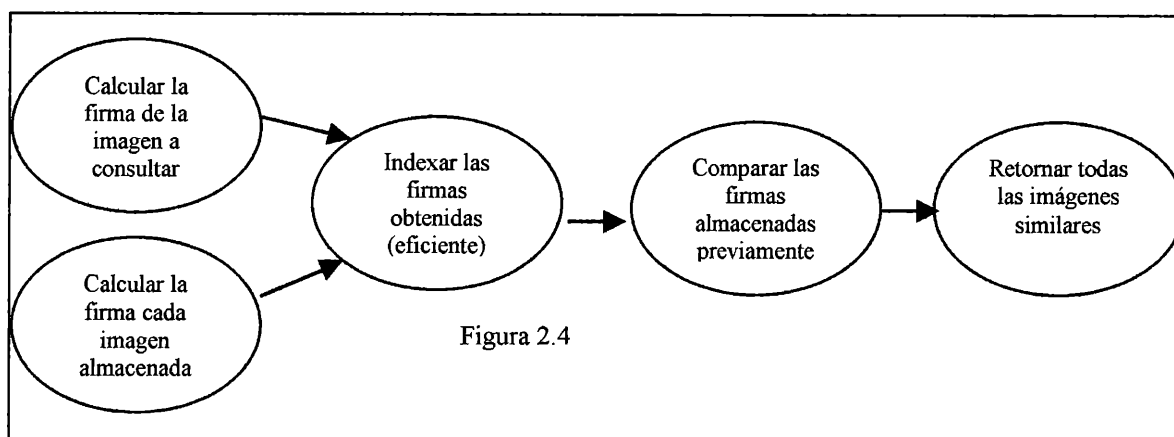
Una solución al problema mencionado consiste en definir firmas independientes sobre cada atributo de la imagen (color, textura, forma, etc) y luego combinarlas para obtener mejores resultados. Es decir, calcular para cada imagen a comparar una firma que tenga en cuenta la textura, otra que considere el atributo color, etc, para luego combinar las mismas obteniendo la firma definitiva para la imagen.

2.2.3. CONSULTAS POR CONTENIDO:

El problema de recuperación de imágenes por contenido consiste en calcular una firma para cada imagen, para luego mapearlas a una estructura d-dimensional en algún espacio de almacenamiento, y construir un índice sobre esta estructura para trabajar con ella más rápidamente.

Posteriormente se define una función de distancia apropiada (ej: euclídea) para cada una de las imágenes. Luego se utiliza el índice para encontrar eficientemente todas las imágenes coincidentes en su firma con la imagen original, obteniendo así el resultado de la consulta.

Un esquema de los pasos explicados anteriormente está representado en la Figura 2.4:



2.3.APLICACIONES DEL PROCESAMIENTO DE IMÁGENES:

Desde la década del 50, el procesamiento de imágenes digitales ha estado en continua evolución gracias a los avances tecnológicos (hardware y software) que se han estado desarrollando. Existe un amplio espectro de aplicaciones que utilizan la tecnología de imagen digital, entre ellas tenemos:

- ❖ Investigaciones biológicas: se realizan mejorados de imágenes, para resaltar rasgos que no son evidentes o claros en la imagen original. Análisis de huesos, tejidos y células. Análisis de ADN.
- ❖ Defensa/inteligencia militar: reconocimiento e interpretación de fotos, en particular en imágenes satelitales para detección de objetos y amenazas militares (aviones, embarcaciones proyectiles, etc). También en este campo se encuentra el reconocimiento y rastreo de objetivos en tiempo real, para bombas inteligentes y sistemas de misiles dirigidos.
- ❖ Procesamiento de documentos: conversión de documentos en papel a una forma de imagen digital (escanear, comprimir, archivar y transmitir documentos), como así

también el reconocimiento automático de caracteres impresos como cheques bancarios y formularios.

- ❖ Diagnóstico médico de imágenes: incluyendo rayos X, tomografías computadas, angiografías
- ❖ Sensado remoto/ recursos terrestres: análisis de suelo a través de imágenes satelitales, con el objetivo de medir rasgos de vegetación tales como contenido de agua, temperatura, características de absorción, etc.

2.4.EXTRACCIÓN DE CARACTERÍSTICAS, CODIFICACIÓN, FIRMA DIGITAL

2.4.1. NECESIDAD DE LA CODIFICACIÓN:

El volumen de datos requeridos para el almacenamiento (y procesamiento) de una imagen, hace conveniente trabajar sobre una *codificación* de la imagen, tratando de trabajar sobre un conjunto mínimo de datos que respete (y permita reconstruir) las características más importantes de la imagen.

Asimismo en ocasiones la codificación permite eliminar información redundante y se puede trabajar en el mejoramiento y análisis de la imagen, directamente sobre la representación codificada de la misma. [Gon96].

En el primer esquema de etapas para el procesamiento de una imagen la codificación se puede ubicar en la etapa de preprocesamiento.

Ejemplos de codificación son las técnicas de Huffman, los histogramas por nivel de gris o color, la transformada de Fourier, la representación mediante Wavelets, etc.

Naturalmente el nivel de compresión respecto del tamaño original de la imagen puede estar asociada con una pérdida relativa de información.

Siempre interesa que la codificación admita inversa (es decir recuperar la imagen original o una aproximación con el menor error posible a la imagen original). En ocasiones también es importante la invariancia de la codificación a modificaciones de la imagen tales como traslaciones, cambios de color, cambios de escala o cambios de textura.

2.4.2. ANÁLISIS DE CARACTERÍSTICAS DE UNA IMAGEN MEDIANTE SU FIRMA DIGITAL:

El objetivo de calcular una firma para la imagen es poder compactar una o más características de la misma.

Luego esta firma puede utilizarse para poder comparar atributos entre imágenes. Para ello, se pueden utilizar diversas métricas que permitan evaluar ciertos atributos entre dos imágenes a partir de sus firmas digitales, y luego establecer un nivel de aproximación entre dichos atributos.

En este trabajo, se evalúa la transformada de wavelet como firma digital de una imagen y se realiza el análisis de similitud basándose en la distancia euclídea entre ambas transformadas o, más precisamente, entre ambos conjuntos de transformadas, dado que la firma digital de una imagen estará representada por el conjunto de transformadas aplicadas sobre las ventanas deslizantes de la misma.

El nivel de aproximación entre ambas imágenes está determinado en base a una cota inferior preestablecida para el porcentaje del área similar, según la distancia euclídea entre las transformadas de wavelet de las ventanas deslizantes.

CAPÍTULO 3: CODIFICACIÓN DE IMÁGENES DIGITALES UTILIZANDO LA TRANSFORMADA DE WAVELET

3.1.INTRODUCCIÓN

Las wavelets son funciones que satisfacen ciertos requerimientos matemáticos y son utilizadas como base para la representación de datos o de otras funciones [Cas95][Cod92][Mas94]. Permiten describir una función en términos de un contorno general mas detalles que van de generales a específicos[Fou97]. Sin importar si la función de interés es una imagen, na curva o una superficie, las wavelets ofrecen una técnica elegante para representar los niveles de detalle presentes.

Las wavelets son muy adecuadas para aproximación de datos con variaciones o con discontinuidades abruptas. La idea fundamental detrás de las wavelets es analizar funciones de acuerdo a escalas. En el análisis por wavelets la escala que se utiliza para analizar los datos juega un papel especial.

Los algoritmos que utilizan wavelets procesan los datos a diferentes escalas o resoluciones. Si se observa una señal o función utilizando una “ventana” ancha, no se observan los pequeños detalles; en cambio, si la “ventana” utilizada es angosta, entonces se los puede observar[Cas95].

El procedimiento general del análisis utilizando wavelets es adoptar una función “prototipo”, llamada generalmente wavelet madre. La señal temporal se realiza entonces utilizando dilataciones y traslaciones de dicha función. La señal original se podrá entonces representar como combinación lineal de la función original y de sus trasladadas y dilatadas. Esto se denomina una expansión en wavelets.

La elección de la wavelet “madre” [Cas95] (y de este modo de la base o del marco de wavelets) no es única y depende del tipo de función o de datos a analizar. Una elección adecuada o la eliminación de coeficientes por debajo de un umbral prefijado, hace que las wavelets sean una herramienta excelente, entre otras, para la compresión de datos[Fou97].

3.2.CODIFICACIÓN CON WAVELET

3.2.1.TRANSFORMADA WAVELET DISCRETA

A continuación se presenta en primer lugar la transformada wavelet discreta diádica y a partir de ella la transformada wavelet continua [Nat99][Cas95][Cod92].

Sea $\psi(x) \in L^2(\mathbb{R})$ la wavelet madre que origina, mediante traslaciones y dilataciones, una base ortonormal. Es decir, el conjunto de funciones que forman la base está definido de la siguiente manera:

$$\psi_{m,n}(x) = 2^{-m/2} \psi(2^{-m}x - n) \text{ con } m, n \in \mathbb{Z}$$

A la descomposición originada por este tipo de wavelets se lo denomina descomposición en octavas. El entero m en la escala indica el ancho de la wavelet y el índice n da la localización en el tiempo. En este caso, la wavelet madre está dilatada o reescalada en escalas que son potencias de 2 y trasladadas en traslaciones enteras.

Una de las ventajas de este tipo de bases es que, conociendo el comportamiento de la wavelet madre, es posible conocer el comportamiento de toda la base.

Es importante señalar que no se puede tomar cualquier función como wavelet madre. Por el contrario, dicha función debe verificar lo que se denomina condición de admisibilidad: $\int \psi(x)DX=0$, de esta manera se restringe a las $\psi(x)$ a funciones de decrecimiento rápido. Por otra parte, la wavelet madre debe oscilar. Esta característica, conjuntamente con su decrecimiento rápido, da su nombre a wavelet (ondita). La condición de admisibilidad permite definir la antitransformada de wavelet.

A partir de la wavelet madre es posible definir, para una función de $L^2(R)$, la transformada wavelet discreta de la siguiente manera:

$$f(m,n) = \int \psi_{m,n}(x) f(x)dx.$$

3.2.2.VENTAJAS DE WAVELET PARA EXTRACCIÓN DE CARACTERÍSTICAS:

La transformada de wavelet permite obtener de manera efectiva una representación compacta que explote la estructura de datos. Usando la descomposición en subbandas de wavelet, y almacenando sólo las subbandas más importantes (los coeficientes superiores), se pueden calcular los vectores de características con un tamaño fijo, independientemente de la resolución, tamaño de la imagen y efectos de “dithering” (esfumado) [Nat99][Cod92][Cas95][Mal97].

También las wavelets son robustas con respecto a cambios de intensidad de color, y pueden capturar de manera eficiente información de textura y forma.

Además las transformadas de wavelet simples pueden calcularse en tiempo lineal, permitiendo la implementación de algoritmos rápidos [Nat99].

3.2.3.APLICACIONES

Aunque los wavelets tienen su origen en la teoría de aproximación y en el procesamiento de señales, han sido aplicados a muchos problemas en gráficos de computadoras[Cas95][Cod97][Mal97][Mas94]. Estas aplicaciones a gráficos incluyen edición de imágenes, compresión de imágenes y consultas de imágenes; control automático de nivel de detalle para editar y presentar curvas y superficies; reconstrucción de superficies a partir de contornos; y métodos rápidos para resolver problemas de simulación en animación y en iluminación global.

3.3.HAAR WAVELET 1D, 2D

3.3.1.TRANSFORMADA HAAR WAVELET

La forma mas simple de wavelets son las bases de Haar. A continuación se describirán las transformaciones de wavelets unidimensionales, para luego discutir generalizaciones bidimensionales:

3.3.2.HAAR WAVELET EN UNA DIMENSIÓN

Las bases de Haar son las bases de wavelets mas simples. Primero se discutirá cómo una función unidimensional se puede descomponer usando los wavelets de Haar, y luego se describirán las bases reales de las funciones[Mal97][Fou97].

Para tener una idea de cómo funcionan los wavelets, se introduce un ejemplo simple. Suponiendo una “imagen” unidimensional con una resolución de 4 pixels, con los siguientes valores:

9 7 3 5

Se puede representar esta imagen en las bases de Haar computando una transformación de wavelet. Para hacer esto, primero se obtiene un promedio de los pixels agrupados de a dos para así obtener una nueva imagen con resolución más baja con valores de pixel:

$$8 \ 4$$

Ciertamente algo de información se ha perdido en este proceso. Para recuperar los 4 valores originales de pixels a partir de los dos promedios, se necesita almacenar algunos coeficientes de detalle, que capturan la información perdida.

En el ejemplo, se obtiene 1 para el primer coeficiente de detalle, ya que el promedio computado es 1 menos que 9 y 1 más que 7. Este número nos permite recuperar los primeros dos pixels de la imagen original de 4 pixels. De manera similar, el segundo coeficiente de detalle es -1 , ya que $4+(-1)=3$ y $4-(-1)=5$.

De esta forma se descompuso la imagen original en una versión de resolución mas baja (dos pixels) y un par de coeficientes de detalle. Repitiendo este proceso sobre los promedios obtenidos se obtiene una descomposición completa, como se observa en la siguiente tabla:

Resolución	Promedios	Coeficientes de detalle
4	9 7 3 5	
2	8 4	1 -1
1	6	2

Finalmente, se definirá la transformada de wavelet (o descomposición de wavelet) de la imagen original de 4 pixels para ser el coeficiente simple que representa el promedio general de la imagen original, seguido de los coeficientes de detalle en orden creciente de resolución. De esta forma, para la base unidimensional de Haar, la transformación del wavelet de la imagen original de cuatro pixels está dada por:

$$6 \ 2 \ 1 \ -1$$

La forma de computar la transformada de wavelet, mediante repetidos cálculos de promedios y coeficientes de diferencia, se llama filter bank, y es un proceso que se generaliza a otro tipo de wavelets. Resulta evidente que en este proceso no se ha perdido ni ganado información.

La imagen original tenía 4 coeficientes, como así también la transformada. Nótese también que, dada la transformación, se puede reconstruir la imagen a cualquier resolución mediante la adición y sustracción de los coeficientes detalle de las versiones de menor resolución.

Una ventaja de representar la imagen mediante la transformada de wavelet en lugar de la imagen misma, consiste en que a menudo una gran cantidad de los coeficientes de detalle resulta ser de muy pequeña magnitud. Estos pequeños coeficientes pueden truncarse o removerse de la representación ocasionando sólo pequeños errores en la imagen reconstruida, dando una forma de compresión de imagen con algo de pérdida [Fou97][Cas95].

3.3.3.HAAR WAVELET EN DOS DIMENSIONES

A continuación se generalizan los wavelets de Haar a dos dimensiones. Se considerará cómo llevar a cabo una descomposición de los valores de pixel de un wavelet en una imagen en 2D [Nat99][Nai00].

Para obtener la descomposición standard de una imagen, primero se aplica la transformada de wavelet unidimensional a cada fila de valores de pixel. Esta operación arroja un valor promedio junto con los coeficientes de detalle para cada fila.

Luego se trata a estas filas transformadas como si fueran ellas mismas una imagen, aplicando la transformación unidimensional a cada columna. Los valores resultantes son



todos coeficientes de detalle a excepción de un único coeficiente promedio. El algoritmo siguiente calcula la descomposición standard[Cas95][Nat97][Fou94].

Procedure Descomposición (C: array[1..h,1..w] of real)

Begin

For fila:= 1 to h do

Descomposición (C[fila,1..w])

For columna:= 1 to w do

Descomposición (C[1..h, columna])

End;

Cálculo de Wavelet en 2-dimensiones (descomposición standard)

El segundo tipo de transformación de wavelet bidimensional, la descomposición no standard, alterna operaciones en filas y columnas. Primero se realiza un paso de promedio y diferenciación de a pares horizontal en los valores de pixel en cada fila de la imagen de resultado.

Para completar la transformación se repite el proceso sucesivamente sólo en el cuadrante que contiene los promedios en las dos direcciones.

Procedure PasoDescomposición (C: array[1..h] of real)

Begin

For i := 1 to h/2 do

Begin

C'[i] := (C[2i-1] + C[2i]) sqrt 2

C'[h/2] := (C[2i-1] + C[2i]) sqrt 2

End;

C := C'

End;

Procedure DescomposiciónNoEstandar (C: array[1..h,1..h] of real)

Begin

C := C / h { para normalizar los coeficientes}

While (h > 1) do

Begin

For fila := 1 to h do

PasoDescomposición (C[fila,1..h])

For columna := 1 to h do

PasoDescomposición (C[1..h, columna])

h := h/2

End;

End;

Procedimiento de Wavelet en 2-dimensiones no-standard

3.3.4.NORMALIZACIÓN:

Una propiedad deseable es la normalización, que permitirá equiparar los pesos de los coeficientes de la transformación[Nat97]. Se pueden normalizar los coeficientes

multiplicando cada coeficiente viejo con superíndice j por $2^{-j/2}$. Así, en el ejemplo previo los coeficientes no normalizados

$$6 \quad 2 \quad 1 \quad -1$$

se transforman en los coeficientes normalizados

$$6 \quad 2 \quad 1/\sqrt{2} \quad -1/\sqrt{2}$$

Como alternativa al procedimiento de primero computar los coeficientes no normalizados y luego normalizarlos, se puede incluir la normalización en la descomposición del algoritmo. Los siguientes procedimientos llevan a cabo esta descomposición normalizada:

Procedure Descomposición (C: array[1..h] of real)

Begin

C := C / sqrt h { para normalizar los coeficientes }

While (h > 1) do

Begin

 PasoDescomposición (C[1..h])

 h := h/2

End;

End;

Cálculo de Wavelet usando una descomposición normalizada

3.4. WAVELET COMO FIRMA DIGITAL PARA EL ANÁLISIS DE SIMILITUD DE IMÁGENES

3.4.1. CALCULO DE LA FIRMA DIGITAL UTILIZANDO LA TRANSFORMADA DE WAVELET:

Se utiliza la transformada de wavelet como firma digital de la imagen debido a sus propiedades de captura de color, textura y forma en una única estructura. Esto representa una ventaja con respecto a otras transformadas utilizadas tradicionalmente (Fourier, Se

utiliza la transformada de Haar wavelet para calcular el vector de características de cada imagen[Nat99][Nai00][Jac95][Wan98].

Las representaciones de wavelet se truncan reteniendo sólo los coeficientes de mayor magnitud, para obtener de esta manera representaciones compactas de las características de las imágenes. Algunas propuestas que utilizaron este enfoque para el análisis de imágenes son Jacobs [Jac95], el sistema WBIIS [Wan98], John Smith [Smi97].

La variante de algoritmo utilizada para la descomposición no estandar implementada en este trabajo se describe en el siguiente pseudocódigo:

Procedure ComputeWavelet (I,W,w)

Begin

For i= 1 to w/2 do

For j= 1 to w/2 do

Begin

$A[i,j] := (I[2i-1,2j-1] + I[2i,2j-1] + I[2i-1,2j] + I[2i,2j])/4$

$W[w/2+i,j] := (-I[2i-1,2j-1] + I[2i,2j-1] - I[2i-1,2j] + I[2i,2j])/4$

$W[i,w/2+j] := (-I[2i-1,2j-1] - I[2i,2j-1] + I[2i-1,2j] + I[2i,2j])/4$

$W[w/2+i,w/2+j] := (I[2i-1,2j-1] - I[2i,2j-1] - I[2i-1,2j] + I[2i,2j])/4$

End;

If (w > 2) then

ComputeWavelet(A,W,w/2)

Else

$W[1,1] := A[1,1]$

End;

Cálculo de Wavelet en 2 dimensiones

A continuación se presenta un ejemplo donde se puede observar el proceso de cálculo de la transformada de wavelet utilizada en dos dimensiones para una imagen I de tamaño 4x4 (observación: por simplicidad sólo se considera una componente de color, la extensión para las 3 componentes es trivial).

Paso 1: Realizar promediados y diferencias horizontales y verticales.

2	4	8	2
4	2	6	4
6	5	6	0
3	2	2	4

Imagen

		0	-2
		-1/2	-2
0	0	-1	1
-6/4	0	0	1

Diferencias verticales y horizontales

3	5
4	3

Promedios

Paso 2: Continuar recursivamente hasta obtener un único coeficiente:

3	5
4	3

Imagen

15/4	1/4	0	-2
-1/4	-3/4	-1/2	-2
0	0	-1	1
-6/4	0	0	1

Diferencias verticales y horizontales (transformada de wavelet)

En el ejemplo se puede observar que a los efectos de obtener una firma digital de la imagen más compacta, alcanza con extraer únicamente los coeficientes de nivel superior de la imagen (ej: el cuadrante superior izquierdo de la transformada de wavelet).

Como resultado final, se puede considerar la matriz

15/4	1/4
-1/4	-3/4

como la firma digital de la imagen original, que posteriormente se utilizará para evaluar el grado de similitud entre dos imágenes.

3.4.2.RECUPERACION DE LA IMAGEN MEDIANTE LA TRANSFORMADA INVERSA DE WAVELET:

De la firma digital extraída en el ejemplo anterior, se puede realizar la antitransformada de Haar wavelet para recuperar la imagen original (en caso de utilizar la transformada de wavelet sin descartar ningún coeficiente de detalle) o una aproximación de la imagen original (en caso de utilizar la firma digital compacta de la transformada, luego de descartar los coeficientes de detalle de menor peso). El segundo caso se suele denominar decodificación de la imagen con pérdida, debido a la pérdida de resolución de la imagen original después de descartar los promedios y diferencias que menos afecten a la recuperación de la misma[Fou97][Fou94].

La utilización de la transformada de wavelet descartando los coeficientes detalle de menor peso como firma digital de una imagen es otra de las utilidades de wavelet en el área de compresión de imágenes.

En el ejemplo anterior, se puede obtener la imagen original partiendo de la transformada de wavelet:

15/4	1/4	0	-2
-1/4	-3/4	-1/2	-2
0	0	-1	1
-6/4	0	0	1

transformada de wavelet de la imagen a decodificar

Considerando el proceso inverso, se recuperan los valores de los pixels en base a los promedios y diferencias aplicando recursivamente las siguientes ecuaciones:

$$\begin{aligned}
 I[2i,2j] &:= A[i,j] \ W[w/2+i,j] \ W[i,w/2+j] \ W[w/2+i,w/2+j] \\
 I[2i,2j-1] &:= A[i,j] \ W[w/2+i,j] \ W[i,w/2+j] \ W[w/2+i,w/2+j] \\
 I[2i-1,2j] &:= A[i,j] \ W[w/2+i,j] \ W[i,w/2+j] \ W[w/2+i,w/2+j] \\
 I[2i-1,2j-1] &:= A[i,j] \ W[w/2+i,j] \ W[i,w/2+j] \ W[w/2+i,w/2+j]
 \end{aligned}$$

Donde I es la imagen a recuperar, A es el promedio en cada instancia de la recursión del algoritmo ComputeWavelet presentado previamente, y W es la matriz de coeficientes de wavelet (ver algoritmo Compute Wavelet).

Partiendo del cuadrante superior izquierdo de la transformada se recupera la siguiente imagen :

3	5
4	3

Imagen recuperada del cuadrante superior izquierdo de la transformada de wavelet

Luego usando la imagen calculada como promedio de cada cuadrante de la imagen original, se recupera la misma usando los coeficientes de wavelets de los cuadrantes restantes de la transformada:

		0	-2
		-1/2	-2
0	0	-1	1
-6/4	0	0	1

Coeficientes de wavelets
(diferencias entre los promedios y los pixels a recuperar)

Finalmente se obtiene la imagen original, sin pérdida (debido al uso de la transformada completa de wavelet):

2	4	8	2
4	2	6	4
6	5	6	0
3	2	2	4

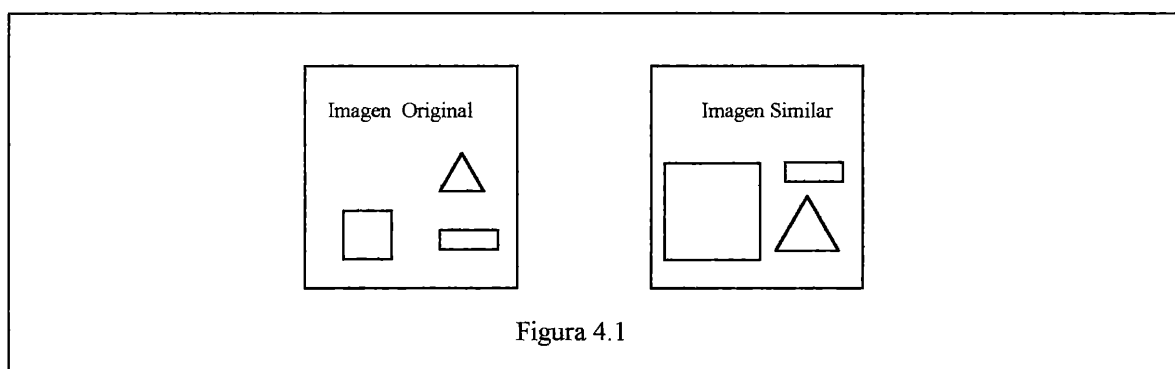
Imagen original recuperada

CAPÍTULO 4: MÚLTIPLES WAVELETS COMO FIRMA DIGITAL

4.1.PROBLEMA DE ESCALADO Y TRASLACIÓN:

El esquema de cálculo de una única firma digital para la imagen completa, usualmente falla cuando las imágenes contienen objetos similares, pero trasladados o en distintos tamaños (cambios de escala)[Nat99][Nai00].

Como ejemplo, se pueden considerar las dos imágenes del siguiente gráfico (Figura 4.1) donde las regiones similares de ambas imágenes comprenden un area significativa. Sin embargo dado que los objetos similares están ubicados en distintos lugares, ambas firmas de Wavelet para las dos imágenes pueden resultar con diferencias importantes.



De esta manera, los algoritmos mencionados anteriormente típicamente fallan con respecto al escalado y traslación de objetos dentro de imágenes, porque una única signature calculada para toda la imagen no puede capturar las propiedades importantes de objetos individuales.

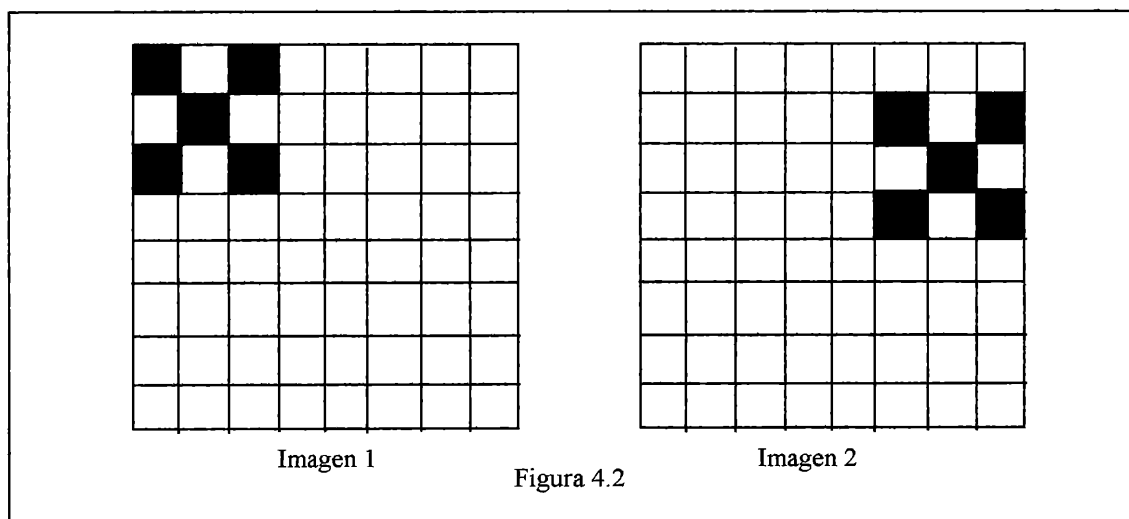
El interés de este trabajo radica en construir un sistema que sea robusto con respecto a cambios de resolución, tamaño y localización, no sólo para la imagen como un todo, también se incluyen los objetos individuales.

Intuitivamente, esto obliga a extender la firma digital más allá de los coeficientes de una wavelet simple.

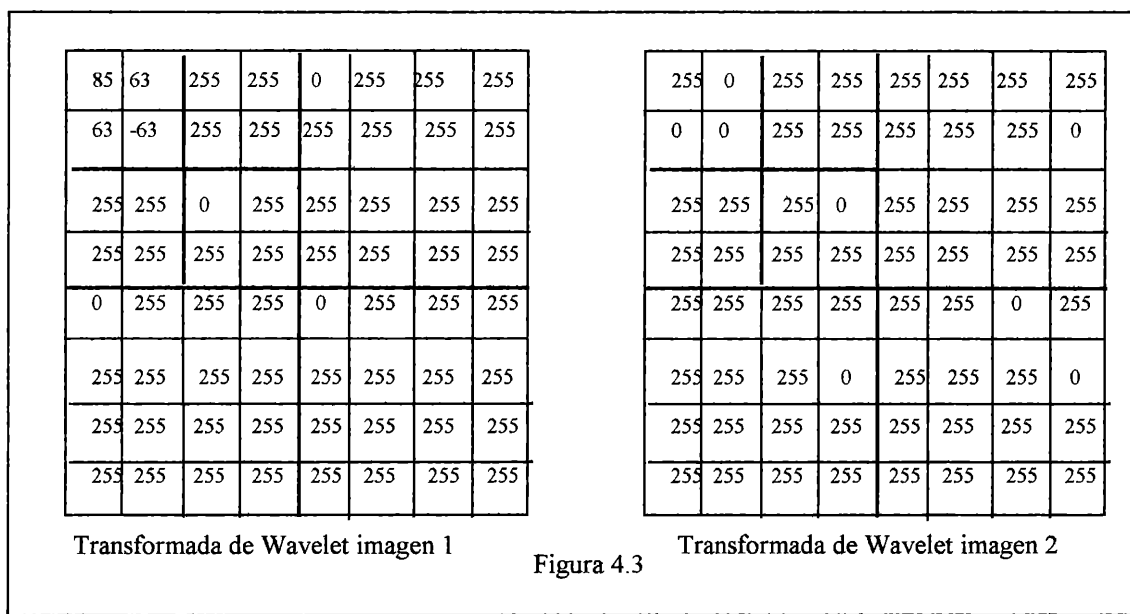
Se propuso un número de estrategias para descomponer una imagen en sus objetos individuales, sin embargo, la extracción de regiones para una imagen es un problema muy complejo de resolver.

Los enfoques que involucran la extracción manual de objetos puede ser demasiado costoso en tiempos de ejecución, por lo que resultan imprácticos para colecciones grandes de imágenes. En consecuencia, la mayoría de las técnicas de segmentación de imágenes se limitan a identificar límites entre regiones, detección de bordes entre objetos, y otros factores, como color, forma, conectividad, etc. Sin embargo, estos esquemas resultan imprecisos en la identificación de objetos y los métodos usados no son robustos con respecto a la granularidad de los objetos. La razón es que la definición de un objeto es muy subjetiva, como resultado, un método simple no puede identificar exitosamente los objetos “correctos” para todas las aplicaciones, y pueden descomponer lo que el usuario percibe como un único objeto en varios objetos menores.

Considerando las siguientes imágenes [Figura 4.2]



En un primer paso, se calcula como firma para cada imagen la transformada de Wavelet tomando la imagen completa, dando como resultado las siguientes matrices que representan la transformada de Wavelet para cada imagen (por una cuestión de simplicidad solo se tendrá en cuenta la componente roja de cada imagen). Figura 4.3



Dado que sólo resultan de interés los coeficientes de mayor peso, se consideran como firmas digitales de ambas imágenes las ventanas ubicadas en el cuadrante superior izquierdo de tamaño 2x2:

85	63
63	-63

Firma digital (imagen 1)

255	0
0	0

Firma digital (imagen 2)

Se observa que el resultado de las firmas de ambas imágenes difiere considerablemente, a pesar de ser similares según las condiciones de similitud presentadas en el capítulo 2.. Esto se debe a que el algoritmo utilizado calcula la firma para cada imagen teniendo en cuenta la imagen completa.

4.1.1.PROBLEMAS DE LA FIRMA DIGITAL DE UNA IMAGEN MEDIANTE WAVELET:

El trabajo propuesto en [Nat99] WALRUS (WaveLet-based Retrieval of User-specified Scenes) se basa en un algoritmo de análisis de similaridad de imágenes para resolver problemas de consulta por contenido.

Dada una imagen Q, WALRUS retorna las imágenes que contienen regiones similares con Q, donde los objetos de Q pueden aparecer en la imagen comparada con cambios de escala, trasladadas o cambios de color. Para poder satisfacer esto, se calculan y comparan firmas digitales en la granularidad de regiones y no de la imagen completa. En lugar de guardar una firma simple por cada imagen, se construye un conjunto de un número variable de firmas digitales para cada imagen, una firma por cada región de la imagen. Se define luego una medida de similaridad entre la imagen Q y la imagen consultada T en términos de la fracción del área de las dos imágenes cubiertas por pares de regiones similares entre Q y T (las regiones que hagan “matching” se determinan basándose en la distancia entre sus firmas digitales).

Se puede observar la mejora de este método con respecto a la utilización de una única firma digital aplicando el algoritmo al ejemplo presentado previamente (Figura 4.2). A estas imágenes se les aplica el algoritmo anterior donde, para simplificar el ejemplo, se utilizan ventanas deslizantes de tamaño 4x4 con desplazamiento 4, obteniendo las siguientes transformadas de wavelet:

0	0	255	255	63	0	255	255	63	0	255	255	63	0	255	255
0	0	255	255	0	0	255	255	0	0	255	255	0	0	255	255
255	255	0	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Transformadas de 4x4 de la imagen 1

63	0	255	255	85	63	255	255	63	0	255	255	63	0	255	255
0	0	255	255	63	-63	255	255	0	0	255	255	0	0	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Transformadas de 4x4 de la imagen 2

Teniendo en cuenta solo el cuadrante superior izquierdo de cada una de las transformadas se tiene:

0	0	63	0	63	0	63	0
0	0	0	0	0	0	0	0

Transformadas de wavelet de la imagen 1

63	0	85	63	63	0	63	0
0	0	63	-63	0	0	0	0

Transformadas de wavelet de la imagen 2

A continuación se procede a agrupar las transformadas similares (según cierta métrica de distancia), obteniendo como firma digital de cada imagen los siguientes conjuntos de transformadas

0	0	63	0	\Rightarrow	0	15
0	0	0	0			

Conjunto de firmas de la imagen 1

85	63	63	0	\Rightarrow	37	15
63	-63	0	0			

Conjunto de firmas de la imagen 2

Finalmente se evalúan las regiones similares y se establece el porcentaje del área coincidente entre ambas imágenes, obteniendo como resultado que ambas imágenes son similares debido a que el área coincidente supera la cota preestablecida.

4.2. CÁLCULO DE WAVELET CON VENTANAS DESLIZANTES DE TAMAÑO VARIABLE:

Para incrementar la precisión en el análisis de similitud de imágenes, [Nat99] propuso el cálculo de Wavelets en dos dimensiones por regiones a través de ventanas deslizantes y de tamaño variable, estableciendo como firma de una imagen un conjunto de firmas correspondientes a las distintas regiones de la misma.

Como resultado, para establecer la similitud entre dos imágenes se analiza el porcentaje de similitudes entre ambos conjuntos de firmas, si el porcentaje de firmas similares supera una cota establecida, se considera que ambas imágenes son similares debido al porcentaje del área coincidente.

Cada imagen se particiona en ventanas deslizantes (que se pueden solapar). Para calcular la firma de cada una de las ventanas solo se utilizan los últimos s coeficientes obtenidos en el cálculo de la transformada de Wavelet.

Evidentemente, almacenar todas las firmas para todas las ventanas generadas tiende a ser muy elevado en términos de espacio y costo de procesamiento. Este overhead se reduce agrupando las ventanas similares de una imagen en un cluster y almacenar sólo una firma representativa de todas las ventanas agrupadas en ese cluster (centroide).

4.3.EXTRACCIÓN DE REGIONES Y SUS FIRMAS DIGITALES:

Se emplea una técnica de extracción de regiones que es relativamente eficiente en costos de ejecución, es independiente del dominio y evita la complejidad de los algoritmos tradicionales de segmentación de imágenes[Zha96][Bec90][Nat99].

Para extraer las regiones de una imagen y calcular sus firmas digitales, se consideran ventanas deslizantes de tamaño variable dentro de la imagen. Para cada ventana, se calcula la firma digital mapeándola en algún espacio d-dimensional, y luego se realiza una etapa de “clustering” de las firmas digitales para agrupar ventanas con características similares (ej: color, textura, forma) en un único cluster. Cada cluster identifica una región de la imagen con valores de pixels relacionados y se usará un cluster representativo como única firma digital de la región (por ejemplo el centroide).

Dado que el modelo de similaridad propuesto compara regiones y las regiones representan ventanas de tamaños variables y en diferentes ubicaciones en la imagen, se eliminan efectivamente los problemas de escalado y traslación, no sólo a nivel de la imagen, también a nivel de objetos.

4.3.1. ESTRATEGIA DE CLUSTERING:

La etapa de agrupamiento “clustering” permitirá representar una región de la imagen compuesta por un conjunto de ventanas de la imagen similares.

Para determinar la similitud entre ventanas de la imagen y decidir agruparlas en una única región se utilizará una métrica de distancia entre un par de ventanas (en nuestras experiencias se utilizó la distancia euclídea entre las firmas digitales de cada ventana). De esta manera cada cluster contiene un conjunto de ventanas similares que juntas definen una región. El centroide del cluster se usará como firma “promedio” de la región.

El objetivo es descomponer una imagen en un número de regiones y obtener una firma digital por cada región.

4.3.2.NECESIDADES DE ALMACENAMIENTO PARA LAS MÚLTIPLES FIRMAS DIGITALES:

Evidentemente, almacenar las firmas digitales para cada ventana deslizante y para cada imagen a consultar puede resultar demasiado costoso en términos de almacenamiento y procesamiento.

Una manera de reducir esta sobrecarga implicó la estrategia de clustering explicada previamente dado que se agrupan ventanas similares dentro de una imagen y se almacena una única firma digital representativa para todas las ventanas en el cluster.

Este trabajo utiliza un árbol de clustering que consiste en una estructura de árbol balanceada, donde por cada entrada de una firma digital correspondiente a una ventana de Wavelet, se recorre el árbol hasta encontrar el cluster mas cercano, y si la ventana resulta similar a la firma digital del cluster (basándose en los centroides de ambas firmas), se considerará que la ventana corresponde a la región representada por ese cluster y la firma digital de la ventana será “absorbida” dentro del cluster, en caso contrario se creará un nuevo cluster, es decir, un nuevo nodo en el árbol de clustering conteniendo como firma digital de la región la firma digital de la ventana. Dependiendo de la complejidad de la imagen dada, el número de clusters puede variar.

Por ejemplo, imágenes simples con características uniformes tienden a poseer muchas ventanas similares, produciendo árboles de clustering pequeños, mientras que imágenes complejas que contienen muchos objetos y transiciones abruptas de color resultarán en árboles de clustering de mayor tamaño, debido al número considerable de ventanas con firmas digitales distintas.

4.3.3. ESTRUCTURA DE CLUSTERING:

En nuestras experiencias se utilizó un árbol binario de clustering[Nai00], donde cada nodo del mismo contiene la firma digital correspondiente al centroide de las firmas digitales de las ventanas que constituyen la región. Además se almacena la cantidad de ventanas deslizantes que hicieron matching con la región y fueron absorbidas por la misma.

A los efectos de poder evaluar en la etapa final del algoritmo cuál fue el área cubierta por la región, teniendo en cuenta que las ventanas deslizantes definen regiones que posiblemente estén solapadas, se utiliza una estructura auxiliar bidimensional con el tamaño coincidente a la imagen a evaluar donde cada posición almacena una lista de clusters a los que pertenece el pixel correspondiente de la imagen. De esta manera de un nodo del árbol de clustering se puede evaluar el factor de peso que tiene la región en la imagen con respecto a la superficie que representa la región en la misma.

4.3.4. CONDICIONES DE SIMILITUD ENTRE IMÁGENES:

Las condiciones que permitirán determinar si dos imágenes son similares se pueden expresar básicamente en dos etapas: Análisis de regiones similares y análisis de imágenes similares[Nat99].

Las regiones generadas en la etapa de clustering para cada imagen produce dos estructuras de clustering que representan la firma digital definitiva de ambas imágenes. El análisis de regiones similares simplemente consiste en encontrar todos los pares de regiones entre ambas estructuras cuyos centroides estén a una distancia dentro de la cota ϵ .

El paso previo de encontrar las regiones similares entre las imágenes a analizar, es decir, los clusters que están dentro de una cota ϵ de distancia entre ambas firmas representadas por un árbol de clustering, retorna todas las regiones de la imagen T que hacen matching con las regiones de la imagen original Q $(Q_1, T_1), \dots, (Q_n, T_n)$. Resulta

evidente que una única región de una de las imágenes puede hacer matching con más de una región de la otra y viceversa.

Se utilizará esta información de regiones coincidentes y la lista de regiones para cada pixel para computar la medición de similaridad entre ambas imágenes como se definió previamente.

La métrica utilizada para calcular el grado de similaridad entre dos imágenes dadas es una donde simplemente se realiza la unión de las regiones coincidentes y se analiza la cantidad de pixels implicados para establecer el porcentaje del área similar considerando similares las imágenes cuyo porcentaje de área similar supera la cota ϵ .

4.4.ALGORITMO DINÁMICO PARA OBTENER LAS MÚLTIPLES FIRMAS DIGITALES DE UNA IMAGEN:

4.4.1.EVALUACIÓN DE WAVELETS UTILIZANDO WAVELETS DE TAMAÑO INFERIOR:

Dada la necesidad de calcular las transformadas de Wavelet para distintas ventanas deslizantes de tamaño variable, resulta conveniente reutilizar el resultado del cómputo de las múltiples ventanas de tamaño N para realizar el cálculo de las múltiples ventanas de tamaño $2*N$ y así sucesivamente.

4.4.2.ALGORITMO DINÁMICO:

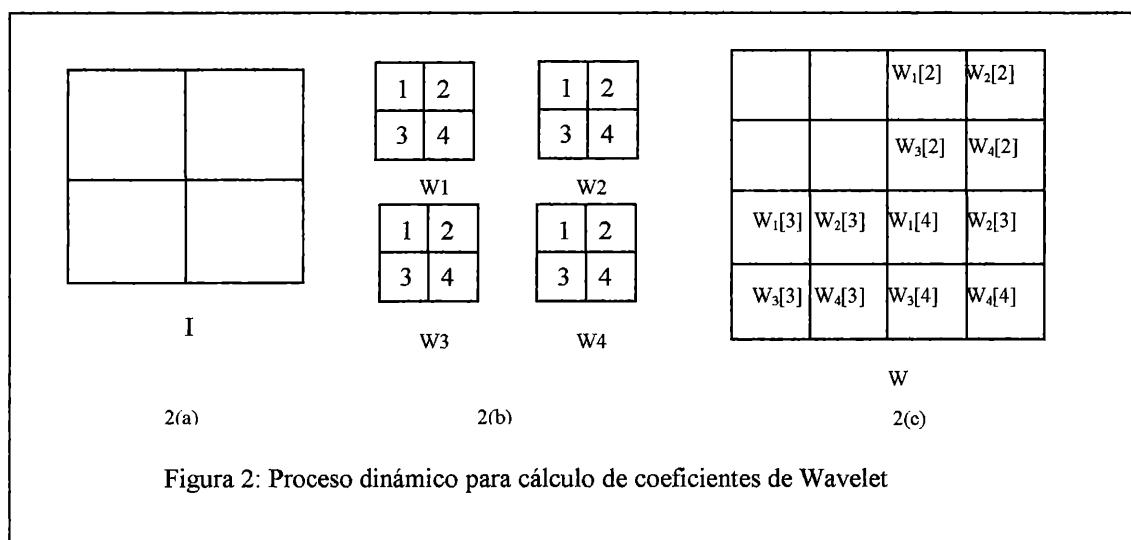
El cálculo de los coeficientes de wavelet para ventanas de tamaño $W \times W$ utilizando ventanas deslizantes de 1 pixel de distancia en una imagen de $N \times N$ implicaría un costo computacional $O(W^2(N-W)^2)$ y considerando que la solución secuencial requiere iterar este

proceso para los distintos tamaños W de ventanas, resulta necesario reducir estos costos de cómputo, [Nat99] propone un algoritmo dinámico que calcula los coeficientes de ventanas de tamaño superior reutilizando el cálculo realizado para las ventanas menores. Asumiendo que las ventanas de tamaño $W/2 \times W/2$ están calculadas, podemos obtener las formas para ventanas de tamaño $W \times W$ usando las firmas de las correspondientes subventanas de tamaño $W/2 \times W/2$.

Este algoritmo obtiene la firma de una ventana I de tamaño $W \times W$ en base a las firmas de sus 4 cuadrantes W_1, W_2, W_3, W_4 (figura 2.a) de la siguiente manera: considerando una nueva descomposición de cada W_i en cuadrantes (figura 2.b), los coeficientes de menor peso correspondientes al 2do, 3er y 4to cuadrante de la ventana I tendrán los valores de los 2dos, 3eros y 4tos cuadrantes de las subventanas W_1, W_2, W_3 y W_4 como se observa en la figura 2.c.

Luego, para calcular el cuadrante superior izquierdo de I , se repite el proceso calculando los coeficientes de ese cuadrante con los coeficientes de los cuadrantes superiores izquierdos de tamaño $W/4 \times W/4$ de W_1, W_2, W_3 y W_4 .

El proceso recursivo termina cuando $W_1[1]$, $W_2[1]$, $W_3[1]$ y $W_4[1]$ contienen un único valor. En este punto los cuatro valores del cuadrante superior de 2×2 de W se obtienen realizando el promediado horizontal y vertical de los 4 valores $W_1[1]$, $W_2[1]$, $W_3[1]$ y $W_4[1]$.



CAPÍTULO 5: PROPUESTA DE PARALELIZACIÓN

5.1 INTRODUCCIÓN A ALGORITMOS Y ARQUITECTURAS PARALELAS

A pesar de los avances en términos de velocidad de procesamiento que se han obtenido en los procesadores, es evidente que hay un límite físico dado por la velocidad de la luz: aunque dentro de un componente electrónico se puedan realizar operaciones muy rápidamente, si se necesita comunicar con otro componente, el tiempo de comunicación de las señales está limitado y se pierde la ganancia de velocidad.

Por lo tanto, la única forma de tratar algunos problemas es la utilización de procesamiento paralelo [Lei92] [Mor94] [Law92]. Si varias operaciones pueden ser ejecutadas simultáneamente, el tiempo total de procesamiento se verá reducido, aún cuando cada una de las operaciones no se lleve a cabo más rápidamente.

Tradicionalmente, el término *supercomputadora* estuvo relacionado de una manera u otra con *procesamiento paralelo*. Al mismo tiempo la especificación de algoritmos ejecutables sobre una arquitectura paralela, ha estado asociada por la *programación concurrente*. [Hoa85] [And91] [Bur93].

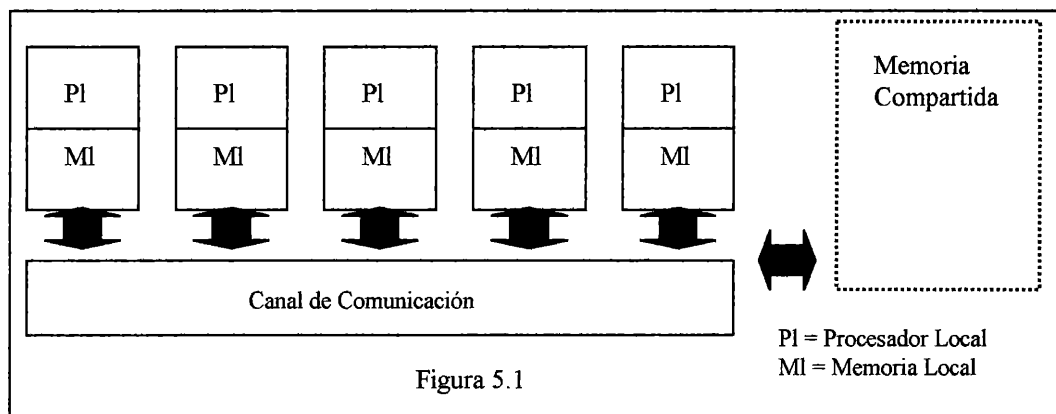
Se han propuesto múltiples formas de organizar las arquitecturas de procesamiento paralelo, e incluso se considera un problema definir qué significa de manera precisa una “arquitectura paralela” [Hwa93]. El problema de definir qué es, y luego dar una taxonomía de las arquitecturas de procesamiento paralelo, se encuentra en la gran cantidad de características que se tienen en una computadora paralela y que no todas son de fácil descripción, comparación y clasificación.

La taxonomía clásica de las arquitecturas de procesamiento establecida inicialmente en [Fly72] se centra en la forma en que se ejecutan las instrucciones sobre los datos:

- **SISD**: **S**ingle **I**nstruction . **S**ingle **D**ata .
- **MISD**: **M**ultiple **I**nstruction . **S**ingle **D**ata .
- **SIMD**: **S**ingle **I**nstruction . **M**ultiple **D**ata.
- **MIMD**: **M**ultiple **I**nstruction . **M**ultiple **D**ata.

Para nuestro interés, nos concentraremos en las arquitecturas genéricamente conocidas como MIMD, tanto de granularidad gruesa (por ejemplo, las redes de máquinas heterogéneas) como de granularidad fina (los hipercubos de pequeños procesadores homogéneos) [Ste96].

La (Figura 5.1) muestra esquemáticamente una arquitectura MIMD genérica, con memoria distribuida. En línea punteada se marca la posibilidad de agregar una memoria compartida.

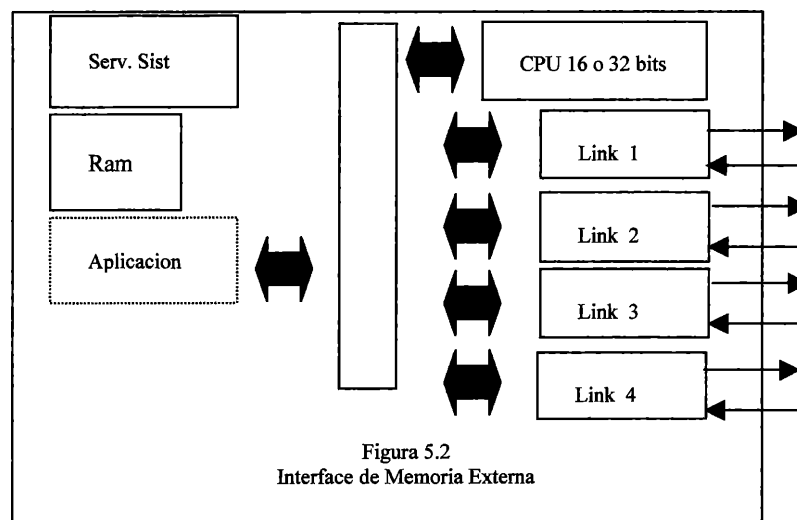


El procesador (P) equivale a la unidad de control (CU) más la unidad de procesamiento (PU). Los streams de datos y de instrucciones (DS y DI) de cada procesador provienen de su propia memoria. A través de los links de comunicaciones más la red de interconexión se pueden enviar mensajes a los demás procesadores. Se podría definir como múltiples SISD interconectadas.

Nótese que en el caso de trabajar con una arquitectura de grano grueso el subsistema de comunicaciones puede ser por ejemplo un bus compartido (local y/o remoto) y la capacidad de cada procesador local equivaler a una Workstation con su propio sistema operativo; en cambio, si se trabaja con un hipercubo de transputers por ejemplo, la capacidad local será mucho menor pero el subsistema de comunicaciones resolverá simultáneamente, por hardware y a muy alta velocidad la vinculación entre los procesos residentes en diferentes procesadores. Estas diferencias hacen particularmente interesante investigar la implantación de algoritmos paralelos clásicos [Kum94] [Ak197] sobre arquitecturas MIMD de diferente granularidad, buscando evaluar parámetros de performance del cómputo paralelo [Gup93] [Gus92] [Bub97].

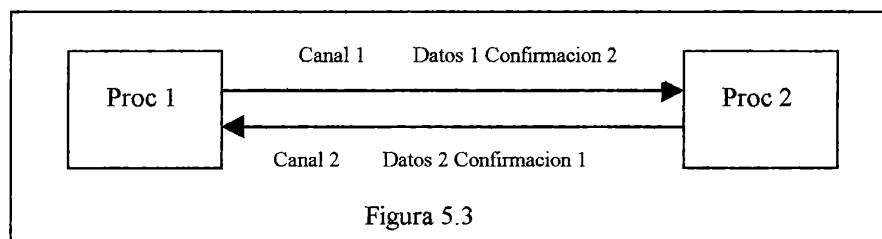
La arquitectura paralela sobre la que se propone la paralelización del algoritmo de múltiples wavelets consiste en una malla de procesadores risc Transputers T805 que permite configurar diversas topologías (en particular el árbol cuaternario de transputers definido para la paralelización del algoritmo dinámico, que calcula la ventana de wavelet en base a los cuadrantes de wavelet como se detalla en este capítulo). A continuación se ofrece una breve descripción del procesador utilizado en este trabajo:

Transputer: Se trata de un procesador tipo RISC (Repertorio de Instrucciones Reducido) constituido por los siguientes bloques: CPU, links, eventos, memoria, interface de memoria externa, reloj y otros servicios (Figura 5.2), pudiendo adicionar una unidad de punto flotante (FPU) o una unidad de procesamiento de disco [CSA90].



Links: Un transputer posee 4 links permitiendo conectarse con un máximo de 4 transputers a través de los mismos. Cada link posee dos controladores de tipo DMA (Acceso Directo a Memoria), uno para entrada y otro para salida. Cada uno de ellos puede leer o escribir en memoria externa o interna sin la intervención de la CPU, permitiendo comunicaciones en paralelo. Cada link provee dos canales, uno en cada dirección, de esta manera un cable transporta datos por un canal y control (ACKs) por el otro, permitiendo la comunicación bidireccional entre 2 transputers (Figura 5.3).

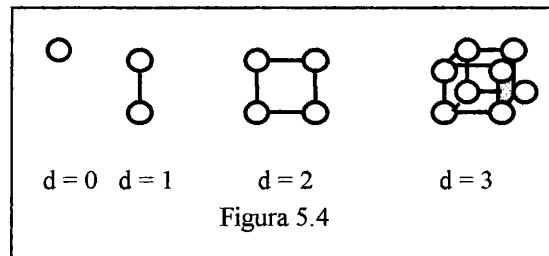
En ciertas aplicaciones es posible la necesidad de comunicación entre procesos alocados en transputers no adyacentes, una solución a esto consiste en la utilización de canales virtuales definidos por software.



Eventos: Las interrupciones son comunicaciones simples sin datos, solo proveen sincronismo. Un pedido de interrupción (EventReq) se asiente (EventAck) si el proceso residente en el transputer esta listo para proceder, también se puede producir una espera de interrupción (EventWaiting).

Interface de memoria externa: Los transputers poseen una interface de memoria externa multiplexada con soporte externo de DMA.

Hipercubo (definición): Un d-cubo está compuesto por 2^d procesadores dispersos en d dimensiones, con 2 procesadores por dimensión y donde cada procesador está conectado a otros d procesadores [figura 5.4]:



Cuando un d-cubo tiene más de tres dimensiones se lo denomina **hipercubo**. La dimensión de un hipercubo está definida de la siguiente manera $d = \log p$ (donde p es la cantidad de procesadores). El **grado** de un hipercubo es igual a la dimensión.

En un hipercubo los procesadores son numerados del 0 al $(2^d - 1)$.

Las características generales de los hipercubos son :

- ❖ Puede ser **definido recursivamente**, esto es, un hipercubo de dimensión cero conecta un solo procesador, y un hipercubo de dimensión uno conecta dos hipercubos de dimensión cero.
- ❖ Es una de las topologías más versátiles y eficientes para el procesamiento paralelo.
- ❖ Puede ser adaptada para tareas de propósito general, así como también para la resolución de problemas específicos (Ejemplo: Transformada de Fourier).
- ❖ Puede simular de manera eficiente cualquier otra topología del mismo tamaño. Un hipercubo de p procesadores puede simular por ejemplo :
 1. Arreglo de p procesadores.
 2. Arbol binario de p procesadores
 3. Malla, Tours, etc.
- ❖ Una de las desventajas principales consiste en que el número de conexiones de cada procesador crece de manera logarítmica con el tamaño de la red. Esto quiere decir que al aumentar la cantidad de procesadores se incrementan las conexiones y el overhead de comunicaciones asociado .

- ❖ El número de procesadores debe ser una potencia de 2; esto constituye una limitación debido al elevado costo de crecimiento de la dimensión del hipercubo.

Cada uno de los procesadores que conforman un hipercubo puede trabajar de dos maneras:

- ✓ Cada procesador trabaja en forma separada, esto se debe a que contiene su propio código de instrucciones con sus propios datos.
- ✓ Los procesadores cooperan entre sí, comunicándose por medio mensajes para compartir sus datos.

Para este trabajo se utilizó un hipercubo de dimensión 5 (constituido por 32 transputers de la serie T800 de 32 bits), cuya estructura se basa en una PC Pentium con una placa Ultra/XLtm que conecta a un grupo de 8 placas VME-XPTM con 4 transputers cada una. La placa Ultra/XLtm dispone de un único transputer **host** sin capacidad de procesamiento, es decir, solo se encarga de las operaciones de entrada/salida del nodo 0.

5.2. MOTIVACIÓN DE PARALELIZACIÓN. PARALELIZACIÓN DE DATOS Y FUNCIONAL.

Si bien se logró reducir el costo computacional del algoritmo de análisis de similitud de imágenes mediante un algoritmo dinámico que permite utilizar el cálculo realizado previamente para ventanas de tamaño inferior, muchas de las aplicaciones que involucran este tipo de procesamiento de imágenes requieren tiempos de respuesta en tiempo real, por lo que resulta necesario incrementar la eficiencia del cómputo de las etapas ya descritas para el algoritmo.

La paralelización sobre una arquitectura multiprocesador es la alternativa natural, debido a las posibilidades de reducir los tiempos de proceso de cada una de las etapas mencionadas.



La complejidad del cálculo involucrado justifica la paralelización y la solución propuesta constituye una combinación de un “pipelining” de multiprocesadores, cada uno de los cuales es una arquitectura paralela homogénea que obtiene los coeficientes de una firma (wavelet). La reusabilidad parcial de los cálculos para las sucesivas firmas obliga al pipelining de estas arquitecturas.

5.3 ARQUITECTURA Y ALGORITMO DE PARALELIZACIÓN

5.3.1 TOPOLOGÍA PROPUESTA DE LA ARQUITECTURA PARALELA

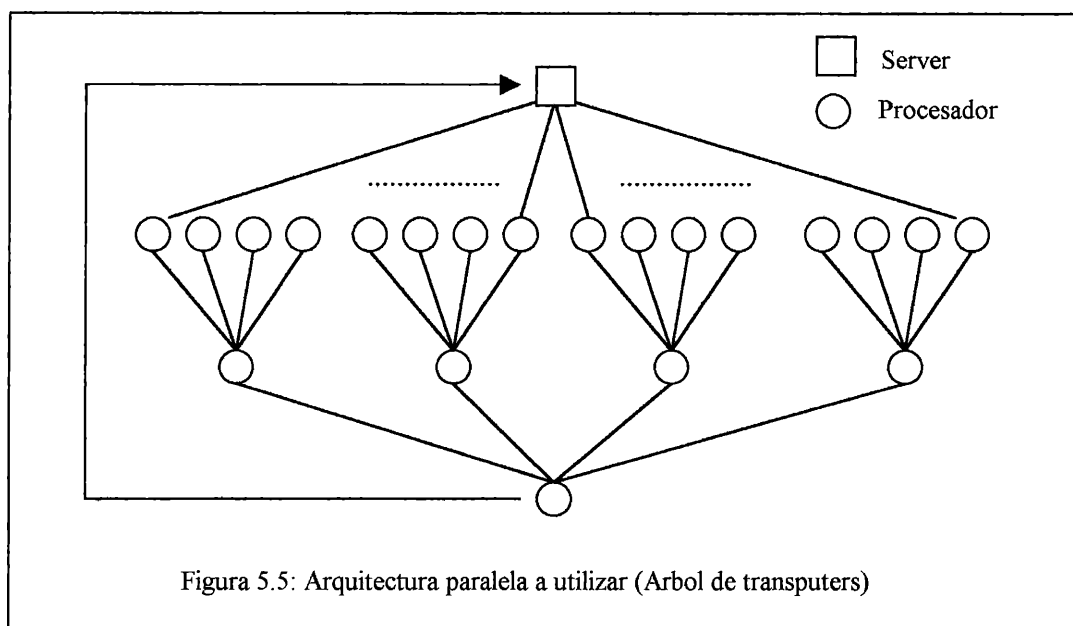
El esquema de la arquitectura paralela a utilizar consiste en un árbol de procesadores (Transputers T805) (Figura 5.5). Este tipo de procesadores resulta adecuado para el modelo de arquitectura planteado debido a su capacidad de conexión física a través de 4 links bidireccionales y memoria local de 4 MB .

Cada procesador se encarga de ejecutar el algoritmo dinámico para calcular una ventana de coeficientes de Wavelet de tamaño W en base a las 4 ventanas recibidas de tamaño $W/2$. Este resultado se envía al procesador asociado del siguiente nivel para que realice el cálculo de la ventana de coeficientes de tamaño $2W$ en base a los resultados de los 4 procesadores asociados del nivel anterior.

Como servidor de las ventanas de coeficientes de Wavelet ya calculadas y recolector de las ventanas de coeficientes de Wavelet retornadas por el árbol de procesadores, se dispone de un procesador Server S que realiza un broadcast de las ventanas y recibe un conjunto de ventanas de coeficientes como resultado del árbol.

Los resultados recolectados por el Server S servirán para realimentar el árbol de procesadores para poder calcular ventanas de tamaño superior. De esta manera el proceso de cálculo de las múltiples ventanas deslizante de tamaño variable consistirá en

un “Pipeline” que alimentará al árbol de procesadores hasta calcular las ventanas deslizantes del tamaño máximo especificado.



5.3.2. ALGORITMO PARALELO:

El algoritmo para obtener la firma digital de una imagen como un árbol balanceado de regiones correspondientes a las firmas digitales de las ventanas de tamaño variable de la imagen, inicia el proceso realizando un broadcast de los valores de pixels individuales como ventanas de coeficientes de tamaño 1x1, de esta manera los procesadores del primer nivel del árbol realizan el cálculo de las ventanas de coeficientes de tamaño 2x2, los procesadores del segundo nivel obtienen las ventanas de coeficientes de tamaño 4x4, etc. El nodo del último nivel del árbol genera su ventana de coeficientes de tamaño N basándose en las cuatro ventanas de tamaño N/2 recibidas y enviará el conjunto de ventanas obtenidas al nodo S, que se utiliza para realimentar el árbol.

Teniendo en cuenta la descripción del algoritmo se puede identificar las siguientes tareas: nodo Server, nodos del medio y nodo final.

El nodo Server tiene la responsabilidad de servir los coeficientes de Wavelets ya calculados, recibir las ventanas de coeficientes resultantes del árbol y generar el árbol de clusters.

Los nodos intermedios del árbol reciben 4 ventanas de coeficientes de tamaño $N/2$, ejecutan el algoritmo dinámico para generar la ventana de coeficientes correspondiente de tamaño N y retornan el conjunto de ventanas de tamaño $N/2$ y la nueva ventana de tamaño N al nodo del siguiente nivel.

Finalmente el nodo ubicado en el último nivel envía al server el conjunto de ventanas resultantes de este paso.

En las siguientes figuras se puede observar una breve descripción de los algoritmos mencionados anteriormente.

Procedure cálculoFirmaDigital (Firma: ArbolBalanceado)

Begin

$w := 1$;

While ($w \leq$ tamaño Máximo) **do**

While (hay ventanas de tamaño w) **do**

 Hacer broadcast de M ventanas de tamaño W al primer nivel de procesadores;

If (hay resultado del último nodo del árbol de procesadores) **then**

 Recibir lista de ventanas de coeficientes del nodo del último nivel;

 Realizar proceso de clustering de la lista de ventanas (Firma);

End If

End While

$w := w * 2$; */* avanza al siguiente tamaño de ventanas deslizantes */*

End While

End;

Figura 5.6: Proceso del Server
(Broadcast, recolección y clustering de ventanas de coeficientes)

Procedure cálculoDinámicoVentanaDeCoeficientes

Begin

While True **do**

 Recibir listas de resultados y 4 ventanas de los nodos del nivel anterior

 ($L1, L2, L3, L4, W1, W2, W3, W4$);

 CálculoDinámicoVentanaDeCoeficientes ($W1, W2, W3, W4, W$);

 Enviar a nodo del nivel inferior lista y resultado ($[L1, L2, L3, L4, W1, W2, W3, W4], W$);

End;

End;

Figura 5.7: Proceso de los nodos del árbol de procesadores
(Algoritmo dinámico para el cálculo de una ventana de coeficientes)

CAPÍTULO 6: RESULTADOS

6.1.IMÁGENES ESTUDIADAS

Para poder evaluar la resistencia del algoritmo en el análisis de similitud frente a cambios de escala (zoom), traslaciones y cambios de color, se planteó un conjunto de imágenes BMP de tamaño cuadrado $N \times N$ (con N potencia de 2), donde ciertos subconjuntos poseen objetos similares con distintos tamaños y trasladados, (Ver imágenes en resultados obtenidos). Debido a las restricciones de memoria existente en los transputers, la prueba se limitó a imágenes simples, sin utilizar el algoritmo con imágenes de gran tamaño y complejas (debido a las necesidades de almacenamiento de las estructuras utilizadas y el tamaño de los árboles de clustering). De todas maneras los resultados obtenidos en las pruebas permitirán concluir la eficacia del algoritmo.

6.2.PARÁMETROS DEL ALGORITMO

La interfaz permite parametrizar el algoritmo estableciendo cotas de tamaño, porcentajes requeridos de similitud y límites de las ventanas deslizantes. Evidentemente, estos parámetros influyen en los resultados obtenidos (que permiten definir valores óptimos de una manera empírica).

Las pruebas realizadas para evaluar la eficacia del algoritmo utilizaron los siguientes valores:

- ❖ **Tamaño máximo de la transformada de Wavelet:** Hace referencia a la cantidad de ventanas deslizantes en las que será dividida la imagen que se estudia. En las pruebas realizadas se utilizaron tamaños 4, 8, 16 y 32 para las ventanas deslizantes.
- ❖ **Tamaño de la firma de la imagen:** Representa la cantidad de coeficientes de la transformada de wavelet utilizadas como firma de la imagen. Este parámetro debe ser

potencia de 2 e influye en la precisión de información acerca de la ventana de la imagen calculada.

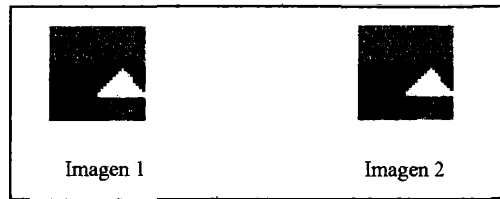
- ❖ **Desplazamiento:** representa la cantidad de pixels de distancia entre dos ventanas adyacentes. Se probó con un conjunto de valores diferentes tales como 2, 4 y 8. Los resultados de las pruebas que se presentan a continuación utilizaron un desplazamiento de 2 pixels para permitir una mayor precisión frente a traslaciones de objetos.
- ❖ **Cota cluster:** representa el valor de cota inferior para determinar en que nodo del árbol de clustering se inserta la nueva firma, en caso que la firma no esté dentro de la cota con ningún nodo del árbol, se crea un nuevo nodo en el mismo conteniendo como firma digital característica la nueva firma. Este parámetro afectará principalmente al tamaño del árbol de Clustering y la precisión de los clusters, es decir, si se utiliza una cota de cluster muy grande, muchas firmas (a pesar de diferir significativamente) se considerarán pertenecientes al mismo cluster formando una firma imprecisa para la región, debido al promediado de las firmas individuales. Por otro lado, si se utiliza una cota demasiado pequeña, la mayoría de las firmas se considerarán como clusters individuales provocando un árbol de clustering demasiado grande.
En las pruebas realizadas se utilizaron 3 valores representando los extremos más exigentes (0.2), menos exigentes (2.0) y un nivel de exigencia intermedio (0.5).
- ❖ **Porcentaje de similitud:** representa la cota que debe superar el porcentaje del área coincidente entre las imágenes comparadas para decidir si son similares o no. En general las pruebas utilizaron porcentajes superiores al 80, 90 y 99% para considerar que dos imágenes son similares.

6.3.RESULTADOS OBTENIDOS

Se realizaron un total de 240 pruebas. A continuación se presenta los resultados que se obtuvieron en algunas de las pruebas realizadas. Estas pruebas se orientan a evaluar imágenes similares con objetos trasladados, cambiados de color, o escalados (zoom). Todos los resultados están expresados en porcentaje.

❖ Prueba 1:

Imágenes a comparar:



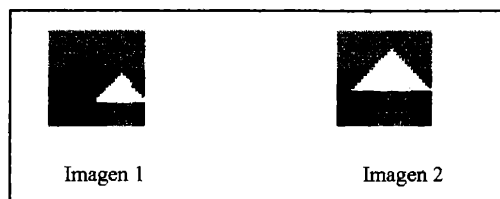
Descripción de las imágenes a evaluar: las imágenes en este caso son iguales. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	100	100	100
8	100	100	100
16	100	100	100
32	100	100	100

❖ Prueba 2:

Imágenes a comparar:



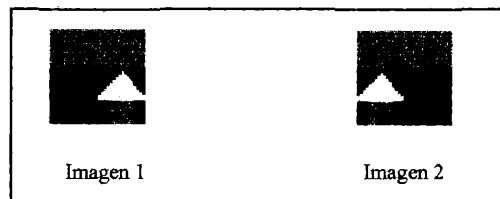
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia de que la casa ubicada en la imagen 2 es mayor al de la casa en la imagen 1 (ZOOM). Lo que se evalúa en este caso es como funciona el algoritmo con imágenes escaladas. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

CC \ TWM	0.02	0.5	2.0
4	99	94	99
8	100	100	99
16	100	100	99
32	100	100	100

❖ Prueba 3:

Imágenes a comparar:



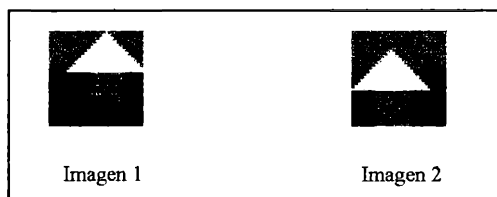
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la casa en la imagen 2 está ubicada sobre la esquina inferior izquierda, mientras que la casa en la imagen 1 está en la esquina inferior derecha. En esta prueba hay una traslación sobre el eje de coordenadas X. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes trasladadas. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	100	100	99
8	100	100	100
16	100	100	99
32	100	100	100

❖ Prueba 4:

Imágenes a comparar:



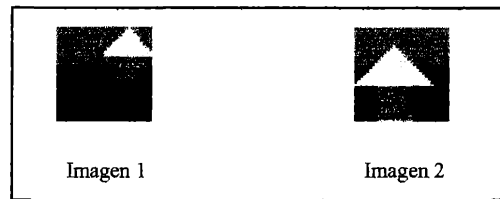
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la casa en la imagen 2 está ubicada sobre la esquina inferior izquierda, mientras que la casa en la imagen 1 está en la esquina superior derecha. En esta prueba hay una traslación sobre los ejes de coordenadas X e Y. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes trasladadas. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similitud) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	99	100	98
8	98	100	98
16	100	100	100
32	100	100	100

❖ Prueba 5:

Imágenes a comparar:



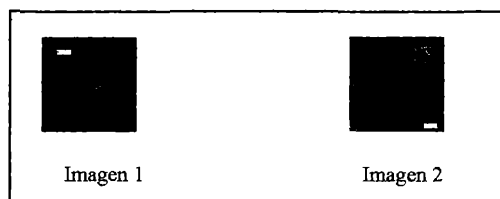
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la casa en la imagen 2 está ubicada sobre la esquina inferior izquierda, mientras que la casa en la imagen 1 está en la esquina superior derecha y es de un tamaño menor. En esta prueba hay una traslación sobre los ejes de coordenadas X e Y. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes trasladadas y escaladas (ZOOM) al mismo tiempo. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	98	98	100
8	99	100	99
16	100	100	100
32	100	100	100

❖ Prueba 6:

Imágenes a comparar:



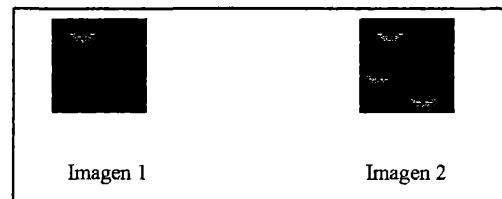
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que los objetos de la imagen 2 se encuentran en diferentes posiciones a los de la imagen 1. En esta prueba hay una traslación sobre los ejes de coordenadas X e Y. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes trasladadas. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	98	98	100
8	99	100	99
16	100	100	100
32	100	100	100

Prueba 7:

Imágenes a comparar:



Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la imagen 2 contiene tres palomas en lugar de una. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	97	99	100
8	96	99	100
16	100	100	100
32	100	100	100

❖ Prueba 8:

Imágenes a comparar:



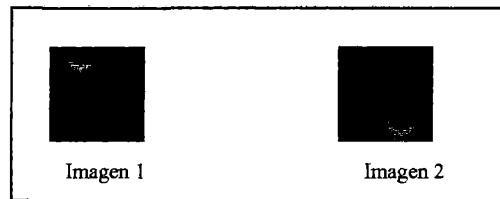
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la imagen 2 contiene la paloma en la esquina inferior izquierda y en la imagen 1 está ubicada en la esquina superior derecha. En esta prueba hay una traslación sobre los ejes de coordenadas X e Y. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes trasladadas. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	96	94	99
8	99	100	100
16	99	100	100
32	99	100	100

❖ Prueba 9:

Imágenes a comparar:



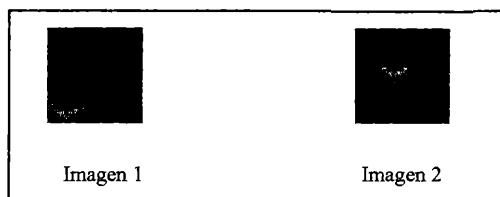
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la imagen 2 contiene la paloma en la esquina inferior derecha y en la imagen 1 está ubicada en la esquina superior izquierda. En esta prueba hay una traslación sobre los ejes de coordenadas X e Y. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes trasladadas. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	97	99	100
8	99	98	100
16	100	100	100
32	100	100	100

❖ Prueba 10:

Imágenes a comparar:



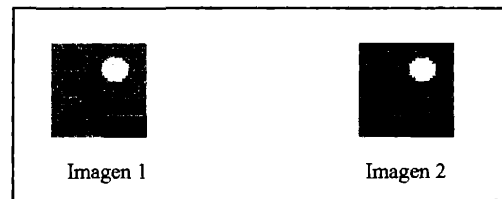
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la imagen 2 contiene la paloma en el centro de la imagen y en la imagen 1 está ubicada en la esquina inferior izquierda. En esta prueba hay una traslación sobre los ejes de coordenadas X e Y. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes trasladadas. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	98	99	87
8	99	100	100
16	100	100	100
32	100	100	100

❖ Prueba 11:

Imágenes a comparar:



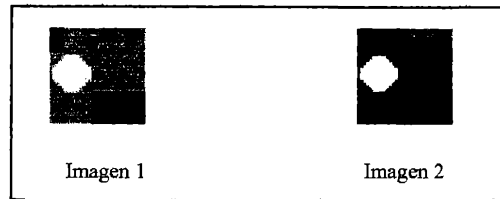
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la imagen 2 contiene el fondo de la imagen en naranja y en la imagen 1 el fondo es azul. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes que cambia el fondo. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

CC	0.02	0.5	2.0
TWM			
4	100	100	100
8	100	100	100
16	100	100	100
32	100	100	100

❖ Prueba 12:

Imágenes a comparar:



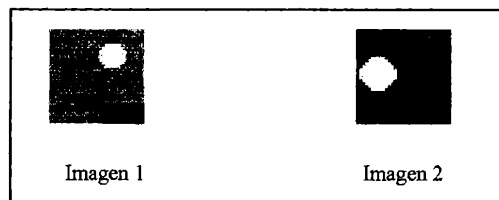
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la imagen 2 contiene el fondo de la imagen en verde y en la imagen 1 el fondo es naranja. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes que cambia el fondo. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	100	100	99
8	100	100	100
16	100	100	100
32	100	100	100

❖ Prueba 13:

Imágenes a comparar:



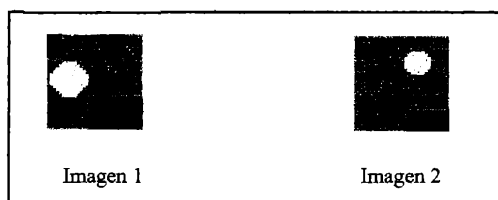
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la imagen 2 contiene el fondo de la imagen en verde con los objetos de un tamaño mayor, y en la imagen 1 el fondo es naranja. Además los objetos se encuentran trasladados sobre los ejes de coordenadas X e Y. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes en las que hay cambios en el color de fondo, cambios de escala (zoom) y traslaciones. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	100	100	99
8	100	98	98
16	100	100	99
32	100	100	100

❖ Prueba 14:

Imágenes a comparar:



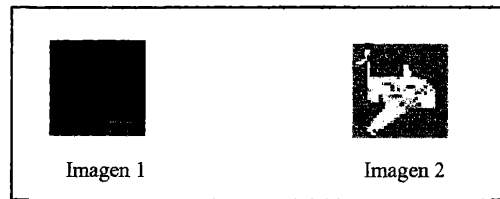
Descripción de las imágenes a evaluar: en este caso las imágenes tienen los mismos objetos, pero con la diferencia que la imagen 2 los ubica en diferentes posiciones y con diferente tamaño. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes que tienen objetos trasladados y escalados (zoom). En esta prueba hay una traslación sobre los ejes de coordenadas X e Y. El resultado esperado es que las imágenes sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	100	100	100
8	100	100	100
16	100	100	100
32	100	100	100

❖ Prueba 15:

Imágenes a comparar:



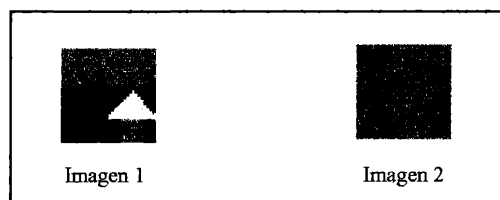
Descripción de las imágenes a evaluar: en este caso las imágenes son totalmente distintas (se nota que el fondo de ambas imágenes es similar). Lo que se evalúa en este caso es como funciona el algoritmo con imágenes diferentes. El resultado esperado es que las imágenes no sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

CC \ TWM	0.02	0.5	2.0
4	0.91	2.6	4.4
8	0.90	3.14	6.5
16	0.90	3.4	6.2
32	0.90	3.4	6.2

❖ Prueba 16:

Imágenes a comparar:



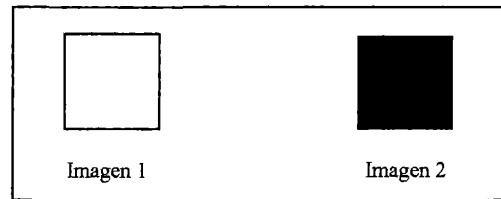
Descripción de las imágenes a evaluar: en este caso las imágenes son totalmente distintas. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes diferentes. El resultado esperado es que las imágenes no sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	1	23	80
8	16	60	80
16	20	60	80
32	20	60	80

❖ Prueba 17:

Imágenes a comparar:



Descripción de las imágenes a evaluar: en este caso las imágenes son totalmente distintas. Lo que se evalúa en este caso es como funciona el algoritmo con imágenes diferentes. El resultado esperado es que las imágenes no sean consideradas similares (según la cota de similaridad) por el algoritmo.

Resultado obtenido:

TWM \ CC	0.02	0.5	2.0
4	0	0	0
8	0	0	0
16	0	0	0
32	0	0	0

CAPÍTULO 7: CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

7.1.CONCLUSIONES

Se ha presentado un algoritmo de análisis de similitud de imágenes, basado en la obtención de múltiples firmas digitales de las imágenes a comparar y se ha establecido una métrica de similitud.

Se ha probado el reconocimiento del grado de similitud entre imágenes a partir del esquema propuesto y se muestra la eficacia del método para un banco de imágenes con objetos trasladados en sus dos ejes, con cambios de escala y de color.

Se esquematizó una solución paralela, partiendo de una arquitectura multiprocesador homogénea basada en transputers.

7.2.LÍNEAS DE TRABAJO FUTURAS

La obtención de los valores óptimos de parametrización del algoritmo y su posterior aplicación sobre una base de datos de imágenes para el problema de consulta de imágenes por contenido son líneas de investigación futura.

Asimismo, como trabajo futuro interesa investigar los problemas de eficiencia y escalabilidad del algoritmo paralelo, así como su posible migración a otra arquitectura física.

El trabajo está abierto a alternativas de paralelización pensando en arquitecturas con memoria compartida y la paralelización de otras etapas del algoritmo, entre las que se pueden citar las etapas de clustering y del cálculo de las tres componentes de color.



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

REFERENCIAS

- [Akl89] Akl S, "The Design and Analysis of Parallel Algorithms", Prentice-Hall, Inc., 1989.
- [Akl97] Akl S, "Parallel Computation. Models and Methods", Prentice-Hall, Inc., 1997.
- [And91] Andrews G., "Concurrent Programming", The Benjaming/Cummings Publishing Company, 1991.
- [Bec90] Beckmann N., Kriegel H., Schneider R., Seeger B., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proceedings of the ACM SIGMOD, Atlantic City 1990. Pags. 322-331.
- [Bri95] Brinch Hansen, P., "Studies in computational science: Parallel Programming Paradigms", Prentice-Hall, Inc., 1995.
- [Bub97] Bubak M., Funika W., Moscinski J., "Performance Analysis of Parallel Applications under Message Passing Environments", www.icsr.agh.edu.pl/publications/html/perf_full/perf_full.html, 1997
- [Bur93] Burns A., Davies G., "Concurrent Programming", Addison Wesley, 1993.
- [Cas95] Castro L., Castro S., "Wavelets y sus Aplicaciones", En los proceedings del 1er. Congreso Argentino de Ciencias de la Computación, Argentina 1995. Pags. 195-204.
- [Cha88] Chandi K. M., Misra J., "Parallel Program Design. A Foundation", Addison Wesley, 1988.
- [Cod92] Cody Mac A., "The Fast Wavelet Transform, Beyond Fourier Transforms", Dr. Dobb's Journal. April 1992. Pags. 16-28.
- [Cra87] Crane R., "A simplified approach to Image Processing", Prentice Hall, 1987.
- [CSA90] "Transputer Architecture", Computer System Architects, 1990.
- [Deg99] De Giusti L., Tarrío D., Naiouf M., De Giusti A., "Eficiencia y escalabilidad en algoritmos paralelos de cálculo del costo mínimo de caminos en grafos". En los anales del Congreso de las Nuevas Tecnologías Informática 2000. Cuba 2000.
- [Fly72] Flynn M. J., "Some Computer Organizations and Their Effectiveness", IEEE Transactions on Computers, C-21, No 9, September, 1972.
- [Fou94] Fournier A., "Wavelets and their Applications in Computer Graphics". SIGGRAPH 94 Course Notes.
- [Fou97] Fournier N. Castro G., "Algoritmo de Compresión de Imágenes Fijas Utilizando la Transformada de Wavelet", Tesina de Grado Lic. en informática U.N.L.P. 1997.
- [Gon96] González R., Woods R., "Tratamiento digital de imágenes", Addison-Wesley/Diaz de Santos, 1996.
- [Gra95] Graps A., "An Introduction to Wavelets", IEEE Vol. 2, No. 2: Summer 1995, pags. 50-61.
- [Gup93] Gupta A., Kumar V., "Performance properties of large scale parallel systems", Journal of Parallel and Distributed Computing, November 1993.
- [Gus92] Gustafson J. L., "The consequences of fixed time performance measurement", Proceedings of the 25th Hawaii International Conference on System Sciences, Volume III, pp 113-124, 1992.
- [Hoa85] Hoare C. A. R., "Communicating Sequential Processes", Prentice-Hall, 1985.

- [Hus91] Hussain Z., "Digital Image Processing. Practical Applications of Parallel Processing Techniques", Ellis Horwood, 1991.
- [Hwa93] Hwang K., "Advanced Computer Architecture: Paralelism, Scalability, Programability", McGraw-Hill, 1993.
- [IEEE] Colección de "IEEE Transactions on Parallel and Distributed Systems", IEEE.
- [Jac95] Jacobs C., Finkelstein A., Salesin D., "Fast Multiresolution Image Querying", In Proc. Of SIG-Graph 95, pag. 277/286. <http://www.cs.washington.edu/research/projects/grail2/www/projects/query/index.html>.
- [Kay95] Kay D., Levine J., "Graphics File Formats", Mc Graw Hill, 1995.
- [Kum94] Kumar V., Grama A., Gupta A., Karypis G., "Introduction to Parallel Computing. Desing and Analysis of Algorithms", Benjamin/Cummings, 1994.
- [Lei92] F. T. Leighton, "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, 1992.
- [Mal97] Mallet Y., "Classification Using Adaptive Wavelets for Feature Extraction", IEEE, vol. 19, No. 10, October 1997.
- [Mas94] Massopust P., "Fractal Functions, Fractal Surfaces, and Wavelets", Academic Press 1994.
- [Mil98] Miller R., Stout Q. F., "Algorithmic Techniques for Networks of Processors", CRC Handbook of Algorithms and Theory of Computation, M. J. Atallah, ed, 1998.
- [Mor94] Morse F., "Practical Parallel Computing", AP Professional, 1994.
- [Nai99] Naiouf M., Tarrío D., De Giusti L., De Giusti A., " Análisis de tiempo y eficiencia en cálculo paralelo del costo mínimo de caminos en grafos". En los anales del III Workshop Chileno en Sistemas Distribuidos y Paralelismo (WCSDP). Talca (Chile) 1999.
- [Nai00] Naiouf M., Tarrío D., De Giusti L., De Giusti A., " Paralelización del Análisis de Similitud de Imágenes", En los anales del VI Congreso Argentino de Ciencias de la Computación, CACIC 2000, (Ushuaia-Argentina) 2000.
- [Nat99] Natsev A., Rastogi R., Shim K., "WALRUS: A Similarity Retrieval Algorithm for Image Databases", Proceedings of the ACM SIGMOD 1999 Philadelphia. Pags. 395-405.
- [Nig95] Nigam M., Sahni S., "Sorting n^2 Numbers on $n \times n$ Meshes", IEEE Transactions on Parallel and Distributed Systems, Vol. 6, No. 12: Diciembre 1995, Pags. 1221-1225
- [Pit93] Pitas I., "Digital Image Processing Algorithms", Prentice Hall. 1993.
- [Smi97] Smith J. R., "Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis", PhD thesis, Graduate School of Arts and Sciences, Columbia University. 1997. <http://www.ctr.columbia.edu/~jrsmith/publications.html>.
- [Ste96] Steenkiste P., "Network-Based Multicomputers: A Practical Supercomputer Architecture", IEEE Transactions on Parallel and Distributed Systems, Vol. 7, No. 8, August 1996, pp. 861-875
- [Tra90] "Transputer Architecture and Overview. Transputer Technical Specifications. ", Computer System Architects, 1990.

[Wan98] Wang J., Wiederhold G. Firschein O., Wei S., "Content-Based Image Indexing and Searching Using Daubechies' Wavelets", Journal of Digital Libraries (IJODL) 1(4):311-328, 1998.

[Zha96] Zhang T., Ramakrishnan R., Livny M., "BIRCH: An Efficient Data Clustering Method for Very Large Databases", Proceedings of the ACM SIGMOD Conference on Management of Data. Pags. 103-114, Montreal, Canada, Junio 1996



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

DONACION.....	TES
\$.....	0014
Fecha..... 4-10-05	
Inv. E..... Inv. D..... 2098	